Linux based 3G Multimedia Mobile-phone

API Specification

[Multimedia]

Draft 1.0

NEC Corporation

Panasonic Mobile Communication Ltd.

Contents

Preface

This document describes an API specification of Multimedia service for Linux based 3G multimedia mobile-phone. This document is the results of the work of the CE Linux Forum's technical working group. The APIs in this document are based on the technology which is originally the collaborative work by NEC Corporation, Panasonic Mobile Communication Ltd., and NTT DoCoMo, Inc.

1. Camera Control

1.1 Event occurrence notification request

Classification	Camera service ELIB function		
Function name	Request for event occurrence notification	Symbol	Elib_CAM_Evt_Request
Function overview			

Registers events to be reported from the camera control service to the APL.

```
#define D_ELIB_CAM_EVT_ALL_ACK              /* Register all events. */
#define D_ELIB_CAM_EVT_POWER_CTL_ACK        /* Camera on/off control response */
#define D_ELIB_CAM_EVT_ZOOM_CTL_ACK         /* Camera zoom control response */
#define D_ELIB_CAM_EVT_DISP_CTL_ACK         /* Camera image adjustment view control response */
#define D_ELIB_CAM_EVT_REC_MODE_ACK         /* Camera recording mode switching control response */
#define D_ELIB_CAM_EVT_REC_RESO_ACK         /* Recording resolution setting completion notification */
#define D_ELIB_CAM_EVT_BRIGHTNESS_ACK       /* Brightness setting completion notification */
#define D_ELIB_CAM_EVT_COLOR_TONE_ACK       /* Color tone setting completion notification */
#define D_ELIB_CAM_EVT_WB_ACK               /* White balance setting completion notification */
#define D_ELIB_CAM_EVT_PHOTO_MODE_ACK       /* Photographing mode setting completion notification */
#define D_ELIB_CAM_EVT_BURST_SPEED_ACK      /* Burst shot interval setting completion notification */
#define D_ELIB_CAM_EVT_SEQSHTNUM_ACK        /* Burst shot count setting completion notification */
#define D_ELIB_CAM_EVT_ZOOM_INFO            /* Zoom information notification */
#define D_ELIB_CAM_EVT_OVERLAY_SETUP_ACK    /* Overlay setting completion notification */
#define D_ELIB_CAM_EVT_PRIMARY_FLIP_ACK     /* Primary flip completion notification */
#define D_ELIB_CAM_EVT_GET_RESOURCE_ACK     /* MMP resource acquisition completion notification */
#define D_ELIB_CAM_EVT_REL_RESOURCE_ACK     /* MMP resource release completion notification */
#define D_ELIB_CAM_EVT_FLICKER_ACK          /* Flicker suppression setting completion notification */
                                    /* Return value */
#define D_ELIB_CAM_OK                       /* Normal end */
#define D_ELIB_CAM_EVT_SEND_ENTRY_NG        /* Failure in event delivery destination registration */
#define D_ELIB_CAM_PARAM_NG                 /* Parameter error */
#define D_ELIB_CAM_OTHERS_NG                /* Other error */
```

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Evt_Request(unsigned int ap_id, int event_id, MsbFunc func)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
event_id	Int	In	Event reported D_ELIB_CAM_EVT_ALL_ACK : Request for all events D_ELIB_CAM_EVT_POWER_CTL_ACK : Camera on/off control response D_ELIB_CAM_EVT_ZOOM_CTL_ACK : Camera zoom control response D_ELIB_CAM_EVT_DISP_CTL_ACK : Camera image adjustment view control response D_ELIB_CAM_EVT_REC_MODE_ACK : Camera recording mode switching control response D_ELIB_CAM_EVT_REC_RESO_ACK : Recording resolution setting notification D_ELIB_CAM_EVT_BRIGHTNESS_ACK : Brightness setting notification D_ELIB_CAM_EVT_COLOR_TONE_ACK : Color tone setting notification D_ELIB_CAM_EVT_WB_ACK : White balance setting notification D_ELIB_CAM_EVT_PHOTO_MODE_ACK : Photographing mode completion notification D_ELIB_CAM_EVT_BURST_SPEED_ACK : Burst shot interval setting notification D_ELIB_CAM_EVT_SEQSHTNUM_ACK: Burst shot count setting completion notification D_ELIB_CAM_EVT_ZOOM_INFO : Zoom information notification D_ELIB_CAM_EVT_OVERLAY_SETUP_ACK : Overlay setting completion notification D_ELIB_CAM_EVT_PRIMARY_FLIP_ACK : Primary flip completion notification D_ELIB_CAM_EVT_GET_RESOURCE_ACK : MMP resource acquisition completion notification D_ELIB_CAM_EVT_REL_RESOURCE_ACK : MMP resource release completion notification D_ELIB_CAM_EVT_FLICKER_ACK : Flicker suppression setting completion notification * Specify events to be reported, by setting pertinent bits. You can

			specify multiple events.
func	MsbFunc	In	Callback function
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Failure in event delivery destination registration : D_ELIB_CAM_EVT_SEND_ENTRY_NG Parameter error : D_ELIB_CAM_PARAM_NG Other error : D_ELIB_CAM_OTHERS_NG

Remarks

- Events are registered if the corresponding bits are set to ON.
- D_ELIB_CAM_EVT_ALL_ACK is the OR of all event types that can be specified.

```
D_ELIB_CAM_EVT_ALL_ACK =
(    D_ELIB_CAM_EVT_POWER_CTL_ACK      | D_ELIB_CAM_EVT_ZOOM_CTL_ACK
     D_ELIB_CAM_EVT_DISP_CTL_ACK       | D_ELIB_CAM_EVT_REC_MODE_ACK
     D_ELIB_CAM_EVT_DISP_CTL_ACK       | D_ELIB_CAM_EVT_REC_MODE_ACK
     D_ELIB_CAM_EVT_COLOR_TONE_ACK     | D_ELIB_CAM_EVT_WB_ACK
     D_ELIB_CAM_EVT_PHOTO_MODE_ACK     | D_ELIB_CAM_EVT_BURST_SPEED_ACK
     D_ELIB_CAM_EVT_SEQSHTNUM_ACK      |
     D_ELIB_CAM_EVT_ZOOM_INFO          | D_ELIB_CAM_EVT_OVERLAY_SETUP_ACK
     D_ELIB_CAM_EVT_PRIMARY_FLIP_ACK   | D_ELIB_CAM_EVT_GET_RESOURCE_ACK
     D_ELIB_CAM_EVT_REL_RESOURCE_ACK   | D_ELIB_CAM_EVT_FLICKER_ACK
);
```

- Only one callback functions can be registered for each Ap_ID.
- If multiple event occurrence notification requests are issued, the callback function that is registered last will be in effect.
- For information on the callback function types, see Chapter 1.13 "Event Format."

1.2 Cancellation of event occurrence notification

Classification	Camera service ELIB function		
Function name	Cancellation of event occurrence notification	Symbol	Elib_CAM_Evt_Cancel
Function overview			

Cancels the registration of events to be reported from the camera control service to the APL.

```
#define D_ELIB_CAM_EVT_NOTIFY_ALL            /* Cancellation of all events */
#define D_ELIB_CAM_EVT_POWER_CTL_ACK         /* Camera on/off control response */
#define D_ELIB_CAM_EVT_ZOOM_CTL_ACK          /* Camera zoom control response */
#define D_ELIB_CAM_EVT_DISP_CTL_ACK          /* Camera image adjustment view control response */
#define D_ELIB_CAM_EVT_REC_MODE_ACK          /* Camera recording mode switching control response */
#define D_ELIB_CAM_EVT_REC_RESO_ACK          /* Recording resolution setting completion notification */
#define D_ELIB_CAM_EVT_BRIGHTNESS_ACK        /* Brightness setting completion notification */
#define D_ELIB_CAM_EVT_COLOR_TONE_ACK        /* Color tone setting completion notification */
#define D_ELIB_CAM_EVT_WB_ACK                /* White balance setting completion notification */
#define D_ELIB_CAM_EVT_PHOTO_MODE_ACK        /* Photographing mode setting completion notification */
#define D_ELIB_CAM_EVT_BURST_SPEED_ACK       /* Burst shot interval setting completion notification */
#define D_ELIB_CAM_EVT_SEQSHTNUM_ACK         /* Burst shot count setting completion notification */
#define D_ELIB_CAM_EVT_ZOOM_INFO             /* Zoom information notification */
#define D_ELIB_CAM_EVT_OVERLAY_SETUP_ACK     /* Overlay setting completion notification */
#define D_ELIB_CAM_EVT_PRIMARY_FLIP_ACK      /* Primary flip completion notification */
#define D_ELIB_CAM_EVT_GET_RESOURCE_ACK      /* MMP resource acquisition completion notification */
#define D_ELIB_CAM_EVT_REL_RESOURCE_ACK      /* MMP resource release completion notification */
#define D_ELIB_CAM_EVT_FLICKER_ACK           /* Flicker suppression setting completion notification */
                                    /* Return value */
#define D_ELIB_CAM_OK                        /* Normal end */
#define D_ELIB_CAM_EVT_SEND_RELEASE_NG       /* Failure in event delivery destination cancellation */
```

#define D_ELIB_CAM_PARAM_NG /* Parameter error */ #define D_ELIB_CAM_OTHERS_NG /* Other error */			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Evt_Cancel(unsigned int ap_id, int event_id)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
event_id	Int	In	Event to be canceled D_ELIB_CAM_EVT_NOTIFY_ALL : Cancellation of all events D_ELIB_CAM_EVT_POWER_CTL_ACK : Camera on/off control response D_ELIB_CAM_EVT_ZOOM_CTL_ACK : Camera zoom control response D_ELIB_CAM_EVT_DISP_CTL_ACK : Camera image adjustment view control response D_ELIB_CAM_EVT_REC_MODE_ACK : Camera recording mode switching control response D_ELIB_CAM_EVT_REC_RESO_ACK : Recording resolution setting notification D_ELIB_CAM_EVT_BRIGHTNESS_ACK : Brightness setting notification D_ELIB_CAM_EVT_COLOR_TONE_ACK : Color tone setting notification D_ELIB_CAM_EVT_WB_ACK : White balance setting notification D_ELIB_CAM_EVT_PHOTO_MODE_ACK : Photographing mode completion notification D_ELIB_CAM_EVT_BURST_SPEED_ACK : Burst shot interval setting notification D_ELIB_CAM_EVT_SEQSHTNUM_ACK: Burst shot count setting completion notification D_ELIB_CAM_EVT_ZOOM_INFO : Zoom information notification D_ELIB_CAM_EVT_OVERLAY_SETUP_ACK : Overlay setting completion notification D_ELIB_CAM_EVT_PRIMARY_FLIP_ACK : Primary flip completion notification D_ELIB_CAM_EVT_GET_RESOURCE__ACK : MMP resource acquisition completion notification D_ELIB_CAM_EVT_REL_RESOURCE__ACK : MMP resource release completion notification

			D_ELIB_CAM_EVT_FLICKER_ACK : Flicker suppression setting completion notification * Specify events to be canceled, by setting pertinent bits. You can specify multiple events.
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Failure in event delivery destination cancellation : D_ELIB_CAM_EVT_SEND_RELEASE_NG Parameter error : D_ELIB_CAM_PARAM_NG Other error :D_ELIB_CAM_OTHERS_NG
Remarks			
- Events are canceled if the corresponding bits are set to ON.			

1.3 Camera resource acquisition

Classification	Camera service ELIB function		
Function name	Camera resource acquisition	Symbol	Elib_CAM_Get_Resource
Function overview			

Acquires camera device resources and MMP power supply resources required when using camera control service functions from the APL.
Sets default values (in still picture mode) of camera status management information (recording mode type) that will come into effect when the power to the camera is turned on.

```
                    /* Return value */
#define D_ELIB_CAM_OK                 /* Normal end */
#define D_ELIB_CAM_OTHER_APL_GAIN_NG  /* Resource used by other APL. */
#define D_ELIB_CAM_DUPLICATE_NG       /* Duplicate resource acquisition error */
#define D_ELIB_CAM_HANDLER_NG         /* Handler error */
```

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Get_Resource(unsigned int ap_id)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Resource used by other APL: D_ELIB_CAM_OTHER_APL_GAIN_NG Duplicate resource acquisition error : D_ELIB_CAM_DUPLICATE_NG Handler error : D_ELIB_CAM_CAM_HANDLER_NG

Remarks

- Before you can use the camera device, you must acquire resources.
- Once camera resource are acquired, they can be locked from use by other applications.
- If an APL acquires camera resources, it must release the camera resources after finishing the camera control task.
- If camera resources that are requested have already been acquired by another application, the function ends abnormally with a "Resource used by another APL" error.
- If the camera resources to be acquired have already been acquired by this application, the function ends abnormally with a "Duplicate resource acquisition" error.
- When the VH completes MMP resource acquisition, the caller application is notified of an event indicating an MMP resource

acquisition completion notification.
- If the MMP resource acquisition completion notification event reports failure in acquisition, the camera resources that
have been acquired by this function will be forced to be released.
- When the camera resource acquisition request is issued, default values (still picture mode) are
assigned to the camera status management information (recording mode type) within the ELIB camera control service.
- When the camera resource acquisition request is issued, the camera mode dependent default zoom value table within
the ELIB camera control service is configured to be zoom-less.
In addition, the camera configuration table containing various camera settings that will be used when the camera is
switched on are initialized to default values. For the default values, see 1.1.6.
Notes:
- If you use camera control service functions without acquiring camera resources, the control sequence
that is being performed by another application may be damaged. This might lead to a camera operation error.
- If the function cannot acquire camera resources, it cannot enqueue for resource acquisition or perform similar processing.
- Before finishing the camera control task, the application must confirm that all control has ended and release the acquired resources.

1.4 Camera resource release

Classification	Camera service ELIB function		
Function name	Camera resource release	Symbol	Elib_CAM_Free_Resource
Function overview			
Releases camera device resources and MMP power supply resources that were acquired when using camera control service functions from the APL. /* Return value */ #define D_ELIB_CAM_OK /* Normal end */ #define D_ELIB_CAM_FREE_RESOURCE_NG /* No resources to be freed */ #define D_ELIB_CAM_OTHER_APL_RELEASE_NG /* Other application resource release error */ #define D_ELIB_CAM_HANDLER_NG /* Handler error */ #define D_ELIB_CAM_OTHER_CTL_ACTION_NG /* Other control action in progress */ #define D_ELIB_CAM_POWER_ON_NG /* Camera power on */			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Free_Resource(unsigned int ap_id)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK No resources to be freed : D_ELIB_CAM_FREE_RESOURCE_NG Other application resource release error : D_ELIB_CAM_OTHER_APL_RELEASE_NG Handler error : D_ELIB_CAM_HANDLER_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power on : D_ELIB_CAM_POWER_ON_NG
Remarks			

1.5 Camera resource re-acquisition (status restoration)

Classification	Camera service ELIB function		
Function name	Camera resource re-acquisition	Symbol	Elib_CAM_ReGet_Resource
Function overview			

Reacquires camera resources and initialize the camera configuration based on the settings that were saved when the camera resources were temporarily freed.
* This interface can only used from the camera APL.

```
                        /* Return value */
#define D_ELIB_CAM_OK                   /* Normal end */
#define D_ELIB_CAM_OTHER_APL_GAIN_NG    /* Resource used by other APL. */
#define D_ELIB_CAM_DUPLICATE_NG         /* Duplicate resource acquisition error */
#define D_ELIB_CAM_HANDLER_NG           /* Handler error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
```

Include File	Srv_cam.h		
Calling Sequence	int Elib_CAM_ReGet_Resource(unsigned int ap_id)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Resource used by other APL : D_ELIB_CAM_OTHER_APL_GAIN_NG Duplicate resource acquisition error : D_ELIB_CAM_DUPLICATE_NG Handler error : D_ELIB_CAM_CAM_HANDLER_NG Parameter error : D_ELIB_CAM_PARAM_NG

Remarks

- This interface can only be used from the camera APL. If it is called by another application, it returns a parameter error.
- The camera configuration is initialized based on the settings (recording mode, brightness, tone type, photographing mode, and default zoom value) that were saved when the camera resources were temporarily freed.
- If there are no saved settings, this function ordinarily acquires camera resources (initializes the camera configuration based on the default settings).
- Saved settings will be discarded when a camera application acquires or reacquires camera resources.
- Other settings (such as white balance setting) may be saved in non-volatile memory. Such settings may have been changed by other applications

while the camera resources were temporarily freed. Therefore, when reacquiring camera resources, check the camera configuration.
- For information on other operation, see the description of camera resource acquisition (Elib_CAM_Get_Resource).

1.6 Camera resource temporary release (status preservation)

Classification	Camera service ELIB function		
Function name	Temporary camera resource release	Symbol	Elib_CAM_TempFree_Resource
Function overview			

Saves the current camera settings and temporarily frees the camera resources.
* This interface can only used from the camera APL.

```
                    /* Return value */
#define D_ELIB_CAM_OK                 /* Normal end */
#define D_ELIB_CAM_FREE_RESOURCE_NG   /* No resources to be freed */
#define D_ELIB_CAM_OTHER_APL_RELEASE_NG  /* Other application resource release error */
#define D_ELIB_CAM_HANDLER_NG         /* Handler error */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG  /* Other control action in progress */
#define D_ELIB_CAM_POWER_ON_NG        /* Camera power on */
#define D_ELIB_CAM_PARAM_NG           /* Parameter error */
```

Include File	Srv_cam.h		
Calling Sequence	int Elib_CAM_TempFree_Resource(unsigned int ap_id)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK No resources to be freed : D_ELIB_CAM_FREE_RESOURCE_NG Other application resource release error : D_ELIB_CAM_OTHER_APL_RELEASE_NG Handler error : D_ELIB_CAM_HANDLER_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power on : D_ELIB_CAM_POWER_ON_NG Parameter error : D_ELIB_CAM_PARAM_NG

Remarks

- This interface can only be used from the camera APL. If it is called by another application, it returns a parameter error.
- Saves camera settings (recording mode, brightness, tone type, photographing mode, and default zoom value)
that would normally discarded at the time of ordinary camera resource release, and then frees the camera resources.

- The saved camera settings will be restored when the camera resources are reacquired (Elib_CAM_ReGet_Resource).
- Saved settings will be discarded when a camera application acquires or reacquires camera resources.
- Other settings (such as white balance setting) may be saved in non-volatile memory. Such settings may have been changed by other applications
while the camera resources were temporarily freed. Therefore, when reacquiring camera resources, check the camera configuration.
- For information on other operation, see the description of camera resource release (Elib_CAM_Free_Resource).

1.7 Camera on/off control

Classification	Camera service ELIB function		
Function name	Camera on/off control	Symbol	Elib_CAM_Ctrl_Camera_Active
Function overview			

Turns on/off the camera (hardware) of the specified camera type. (Also starts/stops image calling.)

```
#define D_ELIB_CAM_CTL_FRONT_ON          /* Front camera on */
#define D_ELIB_CAM_CTL_SIDE_ON           /* Back camera on */
#define D_ELIB_CAM_CTL_ALL_OFF           /* All cameras off */
                                 /* Return value */
#define D_ELIB_CAM_OK                /* Normal end */
#define D_ELIB_CAM_BREAK_IN_POWEROFF_OK  /* Normal end (camera on/off control interrupt acceptance) */
#define D_ELIB_CAM_HANDLER_NG          /* Handler error */
#define D_ELIB_CAM_PARAM_NG            /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG    /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG   /* Other control action in progress */
#define D_ELIB_CAM_CONDITION_NG          /* Camera status error */
#define D_ELIB_CAM_OTHERS_NG           /* Other error */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Ctrl_Camera_Active(unsigned int ap_id, int active)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
active	int	In	Camera operation type Front camera on : D_ELIB_CAM_CTL_FRONT_ON Back side camera on : D_ELIB_CAM_CTL_SIDE_ON All camerasoff : D_ELIB_CAM_CTL_ALL_OFF
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error : D_ELIB_CAM_HANDLER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera status error : D_ELIB_CAM_CONDITION_NG Other error :D_ELIB_CAM_OTHERS_NG
Remarks			

- Before you can call the camera on/off control function, you need to acquire camera resources. If the function is called without resource acquirement, it does not initiate camera device control, but returns abnormally with a Resources not acquired error.
- If a camera (front/back) on control request is issued when other control action is in progress, the function returns abnormally with an Other control action in progress error.
- When a camera off control request is issued when other control action is in progress, the function ends normally (accepts a camera off control interrupt) and returns control to the caller.
- If an attempt is made to turn on a camera that is already on or to turn off a camera that is already off, the function ends abnormally with a Camera status error.
- When a camera on control (front/back camera on) request is issued, the function initializes the specified camera device.
When the camera device is initialized, the recording mode is set based on the value that was previously specified at the time of camera resource acquisition processing (A-3) or recording mode switching control (B-4).
However, if the camera is switched, the recording mode that was in effect before the switching is inherited to the initialization processing.
- When the camera device is initialized, the initialization is based on the camera configuration table containing various camera settings.
The zoom value is set based on the setting in the camera mode dependent default zoom value table.
- If camera on control action (front/back camera on) is performed when another camera is on, camera switching control action is performed.
- When control action (on, off, or switching) on each camera device is completed, the caller application is notified of an event indicating a camera on/off control response notification.

1.8 Camera zoom control

Classification	Camera control service ELIB function		
Function name	Camera zoom control	Symbol	Elib_CAM_Ctrl_Camera_Zoom
Function overview			

Controls camera zoom ratio switching.
* The number of zoom steps varies depending on the camera type and the image resolution.

```
#define D_ELIB_CAM_TYPE_SIDE            /* Back camera */
#define D_ELIB_CAM_TYPE_FRONT           /* Front camera */
#define D_ELIB_CAM_CTL_STEP_ZOOM_IN     /* 1-step zoom in */
#define D_ELIB_CAM_CTL_STEP_ZOOM_OUT    /* 1-step zoom out */
#define D_ELIB_CAM_CTL_ZOOM_RESET       /* Zoom reset */
                                /* Return value */
#define D_ELIB_CAM_OK                   /* Normal end */
#define D_ELIB_CAM_HANDLER_NG           /* Handler error */
#define D_ELIB_CAM_ZOOM_MAX_NG          /* Zoom maximum error */
#define D_ELIB_CAM_ZOOM_MIN_NG          /* Zoom minimum error */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG  /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG         /* Camera power off */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Ctrl_Camera_Zoom(unsigned int ap_id, int camera_type, int zoom_action)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	int	In	Camera type Front camera : D_ELIB_CAM_TYPE_FRONT Back camera : D_ELIB_CAM_TYPE_SIDE
zoom_action	int	In	Zoom action type 1-step zoom in : D_ELIB_CAM_CTL_STEP_ZOOM_IN 1-step zoom out : D_ELIB_CAM_CTL_STEP_ZOOM_OUT Zoom reset : D_ELIB_CAM_CTL_ZOOM_RESET
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error : D_ELIB_CAM_HANDLER_NG Zoom maximum error : D_ELIB_CAM_ZOOM_MAX_NG Zoom minimum error : D_ELIB_CAM_ZOOM_MIN_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG

			Camera power off : D_ELIB_CAM_POWER_OFF_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

Remarks:

- Before you can call the camera zoom control function, you need to acquire camera resources. If the function is called without resource acquirement, it does not initiate camera device control, but returns abnormally with a Resources not acquired error.
- If an attempt is made to control zooming of a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error.
- If an attempt is made to control zooming of a camera device whose power is off, this function returns abnormally with a Camera power off error.
- If an attempt is made to perform a zoom-in action beyond the maximum number of zoom steps, this function returns abnormally with a Zoom maximum error.
- If an attempt is made to perform a zoom-out action below the minimum number of zoom steps, this function returns abnormally with a Zoom minimum error.
- The zoom increment that will be in effect for each resolution used during 1-step zoom in/out action is managed on a constant table by the camera control service and calculated based on this constant table.
- When the zoom control action on each camera device is completed, the caller application is notified of an event indicating a camera zoom control response notification.

Processing:

1-step zoom in/out

- The next number of zoom steps is calculated based on the acquired current number of zoom steps as well as the maximum/minimum number of zoom steps.
- Zoom control action is performed using the [Middle Layer] camera device parameter setting function, which takes the obtained number of zoom steps.
- Camera device parameter setting results are received from the [Middle Layer] VH via a callback function and the zoom management information is updated with the number of zoom steps that is now included in the new camera device configuration.
- The number of current zoom steps and the total number of zoom steps are assigned to the zoom control response event and the caller application is notified of the control result.
- If the zoom control response event is issued for a 1-step zoom control action, it reports the result of the 1-step zoom control action and the zoom information (the number of current zoom steps and the total number of zoom steps).

Zoom reset:

- The minimum number of zoom steps (0 steps) is set as the next number of zoom steps .
- Zoom control action is performed using the [Middle Layer] camera device parameter setting function, which takes the specified number of zoom steps (0 steps).
- Camera device parameter setting results are received from the [Middle Layer] VH via a callback function and the zoom management information is updated with the number of zoom steps (0 steps) that is now included in the new camera device configuration.

- The number of current zoom steps and the total number of zoom steps are assigned to the zoom control response event and the caller application is notified of the control result.
- If the zoom control response event is issued for zoom reset, it reports the result of the zoom reset control action and the zoom information (the number of current zoom steps and the total number of zoom steps).

1.9 Camera image adjustment view control

Classification	Camera service ELIB function		
Function name	Camera image adjustment view control	Symbol	Elib_CAM_Ctrl_Image_Adjust
Function overview			

Controls the camera image adjustment view that is used when configuring the camera by updating the displayed information.
Notifies the [Middle Layer] VH of the image adjustment mode adjustment data specified by parameters to ensure that the image will be displayed based on the adjustment data.
However, the adjustment data is not saved in non-volatile memory.

```
#define D_ELIB_CAM_TYPE_FRONT              /* Front camera */
#define D_ELIB_CAM_TYPE_SIDE               /* Back camera */
#define D_ELIB_CAM_SNAP                    /* Still picture */
#define D_ELIB_CAM_MOVIE                   /* Moving picture */
#define D_ELIB_CAM_BURST_SNAP              /* Burst shooting */
#define D_ELIB_CAM_CTL_ADJ_RESO            /* Screen adjustment view: Resolution */
#define D_ELIB_CAM_CTL_ADJ_RESO_RST        /* Screen adjustment view: Resolution restoration */
#define D_ELIB_CAM_CTL_ADJ_COL_TONE        /* Image adjustment view: Color tone */
#define D_ELIB_CAM_CTL_ADJ_COL_TONE_RST    /* Image adjustment view: Color tone restoration */
#define D_ELIB_CAM_CTL_ADJ_BRIGHT          /* Screen adjustment view: Brightness */
#define D_ELIB_CAM_CTL_ADJ_BRIGHT_RST      /* Screen adjustment view: Brightness restoration */
#define D_ELIB_CAM_CTL_ADJ_WB              /* Image adjustment view: White balance */
#define D_ELIB_CAM_CTL_ADJ_WB_RST          /* Image adjustment view: White balance restoration */
#define D_ELIB_CAM_CTL_ADJ_PHOT_MODE       /* Image adjustment view: Photographing mode */
#define D_ELIB_CAM_CTL_ADJ_PHOT_MODE_RST   /* Image adjustment view: Photographing mode restoration */
                                   /* Return value */
#define D_ELIB_CAM_OK                      /* Normal end */
#define D_ELIB_CAM_HANDLER_NG              /* Handler error */
#define D_ELIB_CAM_DRIVER_NG               /* Driver error */
#define D_ELIB_CAM_PARAM_NG                /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG      /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG     /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG            /* Camera power off */
#define D_ELIB_CAM_OTHERS_NG               /* Other error */
```

Include	srv_cam.h

File			
Calling Sequence	int Elib_CAM_Ctrl_Image_Adjust(unsigned int ap_id, int camera_type, int record_mode, int adj_mode,int adj_data)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	int	In	Camera type Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE
record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE Burst: D_ELIB_CAM_BURST_SNAP * The recording mode is only effective when resolution restoration is specified for the adjustment mode.
adj_mode	int	In	Screen adjustment mode Resolution adjustment : D_ELIB_CAM_CTL_ADJ_RESO Resolution restoration : D_ELIB_CAM_CTL_ADJ_RESO_RST Color tone adjustment : D_ELIB_CAM_CTL_ADJ_COL_TONE Color tone restoration : D_ELIB_CAM_CTL_ADJ_COL_TONE_RST Brightness : D_ELIB_CAM_CTL_ADJ_BRIGHT Brightness restoration : D_ELIB_CAM_CTL_ADJ_BRIGHT_RST White balance adjustment : D_ELIB_CAM_CTL_ADJ_WB White balance restoration : D_ELIB_CAM_CTL_ADJ_WB_RST Photographing mode adjustment : D_ELIB_CAM_CTL_ADJ_PHOT_MODE Photographing mode restoration : D_ELIB_CAM_CTL_ADJ_PHOT_MODE_RST
adj_data	int	In	Adjustment data Resolution adjustment: VGA :D_ELIB_CAM_RESOLUTION_VGA CIF :D_ELIB_CAM_RESOLUTION_CIF QCIF :D_ELIB_CAM_RESOLUTION_QCIF Sub-QCIF :D_ELIB_CAM_RESOLUTION_SUB_QCIF JAVA :D_ELIB_CAM_RESOLUTION_JAVA UXGA :D_ELIB_CAM_RESOLUTION_UXGA SXGA :D_ELIB_CAM_RESOLUTION_SXGA Color tone adjustment: Ordinary : D_ELIB_CAM_COLOR_TONE_NORMAL

			Sepia : D_ELIB_CAM_COLOR_TONE_SEPIA Black and white : D_ELIB_CAM_COLOR_TONE_MONOCHROME Brightness: Brightness level Level 1 (-2) : D_ELIB_CAM_BRIGHTNESS_LEVEL1 Level 2 (-1) : D_ELIB_CAM_BRIGHTNESS_LEVEL2 Level 3 (0) : D_ELIB_CAM_BRIGHTNESS_LEVEL3 Level 4 (+1) : D_ELIB_CAM_BRIGHTNESS_LEVEL4 Level 5 (+2) : D_ELIB_CAM_BRIGHTNESS_LEVEL5 White balance adjustment : Fine : D_ELIB_CAM_WB_FINE_WEATHER Cloud : D_ELIB_CAM_WB_CLOUDY_WEATHER Auto : D_ELIB_CAM_WB_AUTO Bulb : D_ELIB_CAM_WB_ELECTRIC_LIGHT Photographing mode adjustment: Close-up : D_ELIB_CAM_MODE_CONTACT Portrait : D_ELIB_CAM_MODE_PORTRAIT Scenery : D_ELIB_CAM_MODE_SCENERY (Standard : D_ELIB_CAM_MODE_NORMAL) Night mode : D_ELIB_CAM_MODE_NIGHT OCR : D_ELIB_CAM_MODE_OCR Two-dimensional bar code: D_ELIB_CAM_MODE_BARCODE Sports mode: D_ELIB_CAM_MODE_SPORTS * Adjustment data is invalid if the screen adjustment mode indicates "restoration."
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Driver error : D_ELIB_CAM_DRIVER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off : D_ELIB_CAM_POWER_OFF_NG Other error : :D_ELIB_CAM_OTHERS_NG
Remarks			

Remarks:

- Before you can call the camera image adjustment view control function, you need to acquire camera resources. If the function is called without resource acquirement, it does not initiate camera device control, but returns abnormally with a Resources not acquired error.
- If a request is issued to control the camera image adjustment view of a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error.
- If a request is issued to control the camera image adjustment view of a camera device whose power

is off, this function returns abnormally with a Camera power off error.
- When the image adjustment view control action on each camera device is completed, the caller application is notified of an event indicating a camera image adjustment view control response notification.

Image adjustment mode:
- Specify a data type associated with the desired image adjustment (Resolution/Tone/Brightness/White balance/Photographing mode).
- If image adjustment restoration (Resolution/Tone/Brightness/White balance/Photographing mode) is specified, the image will be adjusted based on the data that is currently contained in non-volatile memory.

Adjustment data:
- Specify adjustment data for each adjustment mode.For information on each type of data, see the description of the setting function relating to the mode.
    C-1 Recording resolution setting (Elib_CAM_Set_Record_Resolution)
    C-2 Brightness setting (Elib_CAM_Set_Brightness)
    C-3 Color tone switching (Elib_CAM_Set_Color_Tone)
    C-4 White balance adjustment setting (Elib_CAM_Set_White_Balance)
    C-5 Photographing mode setting (Elib_CAM_Set_Photo_Mode)

Processing: Adjustment -> Setting completion:
- The [Middle Layer] VH function is called to set control values based on the adjustment mode and adjustment data specified by the [ELIB] camera image adjustment view control function.
- The image adjustment view control request response (using a control response callback function) from the [Middle Layer] VH function is received and the application with the specified ID is notified of the control result.
- A camera image is displayed based on the control value specified by the [Middle Layer] VH function.
- This function only adjusts the image on the camera device. Therefore, to save setting data in non-volatile memory, use the appropriate setting function (group C).

Processing: Adjustment -> Restoration:
- The [Middle Layer] VH function is called to set control values based on the adjustment mode and adjustment data specified by the [ELIB] camera image adjustment view control function.
- The image adjustment view control request response (using a control response callback function) from the [Middle Layer] VH function is received and the application with the specified ID is notified of the control result.
- A camera image is displayed based on the control value specified by the [Middle Layer] VH function.
- To restore the pre-adjustment data without storing the image adjustment data (sent to the camera device) into non-volatile memory, call the [ELIB] camera image adjustment setting view control function in which Adjustment mode: View restoration is specified. The original image is then restored.
- To read data from non-volatile memory, use an appropriate [DD Layer] operation API function.
- The [Middle Layer] VH function is called to set control values based on the read data.
- The image adjustment view control request response (using a control response callback function) from the [Middle Layer] VH function is received and the application with the specified ID is notified of

the control result.
- The camera image is displayed based on the pre-adjustment control values.

Middle layer/DD layer interface:
- [DD Layer] operation API non-volatile memory read function

- This function does not store actual settings in non-volatile memory.

1.10 Camera recording mode switching control

Classification	Camera service ELIB function		
Function name	Camera recording mode switching control	Symbol	Elib_CAM_Ctrl_Record_Mode_Change
Function overview			

When the camera recording mode (Still/Moving/Burst) is changed via the APL when the camera is active (for photographing), this function notifies the camera device of the settings for the pertinent recording mode and updates the camera mode dependent default zoom value table with the new settings.
Sets the camera status management information (recording mode) that will come into effect when the power to the camera is turned on.

```
#define D_ELIB_CAM_TYPE_FRONT          /* Front camera */
#define D_ELIB_CAM_TYPE_SIDE           /* Back camera */
#define D_ELIB_CAM_SNAP                /* Still picture */
#define D_ELIB_CAM_MOVIE               /* Moving picture */
#define D_ELIB_CAM_BURST_SNAP          /* Burst shooting */
                                /* Return value */
#define D_ELIB_CAM_OK                  /* Normal end */
#define D_ELIB_CAM_HANDLER_NG          /* Handler error */
#define D_ELIB_CAM_PARAM_NG            /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG  /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG        /* Camera power off */
#define D_ELIB_CAM_OTHERS_NG           /* Other error */
```

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Ctrl_Record_Mode_Change(unsigned int ap_id, int camera_type, int record_mode)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	int	In	Camera type Front camera : D_ELIB_CAM_TYPE_FRONT Back camera : D_ELIB_CAM_TYPE_SIDE
recode_mode	int	In	Recording mode Still picture : D_ELIB_CAM_SNAP Moving picture : D_ELIB_CAM_MOVIE Burst : D_ELIB_CAM_BURST_SNAP
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error : D_ELIB_CAM_HANDLER_NG

			Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off : D_ELIB_CAM_POWER_OFF_NG Other error :D_ELIB_CAM_OTHERS_NG

Remarks

Remarks:

- Before you can call the camera recording mode switching control function, you need to acquire camera resources. If the function is called without resource acquirement, it does not initiate camera device control, but returns abnormally with a Resources not acquired error.
- If an attempt is made to switch the camera recording mode of a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error.
- If an attempt is made to switch the camera recording mode of a camera device whose power is off, this function returns abnormally with a Camera power off error.

However, if all camera devices are in power-off state, this function only updates the camera status management information (recording mode).

If the camera opposite to the camera specified in this function is in power-on state, the function returns abnormally with a Camera power off error without performing any processing.

- When camera recording mode switching is completed, the caller application is notified of an event indicating a camera recording mode switching control completion notification.

Control timing:

- Timing of camera recording mode switching
- Recording mode setting prior to camera power on control action

Processing:

(1) Camera status management information (recording mode) updating
(2) Settings for each recording mode are read from non-volatile memory.
(3) The camera device is configured based on the read settings.
(4) Zoom values are set based on the camera mode dependent default zoom value table.

Middle layer/DD layer interface

- [DD Layer] non-volatile memory read function

1.11 Overlay setting control

Classification	Camera service ELIB function		
Function name	Overlay setting control	Symbol	Elib_CAM_Ctrl_Overlay_Setup
Function overview			

Sets an overlay based on the specified overlay setting information.

```
#define D_ELIB_CAM_DEV_ID_MAIN_LCD       /* Main LCD device ID */
                                 /* Return value */
#define D_ELIB_CAM_OK                    /* Normal end */
#define D_ELIB_CAM_HANDLER_NG            /* Handler error */
#define D_ELIB_CAM_PARAM_NG              /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG    /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG   /* Other control action in progress */
```

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Ctrl_Overlay_Setup(unsigned int ap_id, HmiVmDevID dev, const HmiVmOverlayParam *param)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
dev	HmiVmDevID	In	Overlay-applicable device ID Main LCD: D_ELIB_CAM_DEV_ID_MAIN_LCD

*param	const HmiVmOverlayParam	In	Overlay setting information structure
			typedef struct HmiVmOverlayParam_t {
			unsigned int surfNum; //Number of resulting surfaces
			HmiVmOverlayParamSurf *surf; // Setting information area for each surface
			HmiBoolean sync; // Synchronous flip flag
			}HmiVmOverlayParam;
			&127; See the remarks for details about the structure.

Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG

Remarks

- Before you can call the overlay setting control function, you need to acquire camera resources. If the function is called without resource acquirement, it does not initiate overlay setting, but returns abnormally with a Resources not acquired error.
- If an attempt is made to set an overlay on a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error.
- The caller application needs to allocate areas for the number of surfaces on which the overlay is to be set. (Areas for up to three surfaces need to be allocated.)
- The surfaces are assigned priorities so that Surf member elements closer to the beginning (first element) are assigned higher priorities.However, graphic surface overlay information always needs to be set in the beginning of element members.
- When overlay setting action is completed, the caller application is notified of an event indicating an overlay setting completion notification.
- For details about the overlay setting information structure, surface information (such as coordinates), and surface combinations, see the detailed description of the HmiVmOverlayParam and HmiVmOverlayParamSurf structures in the Video Handler Interface Specifications.

1.12 Primary flip control

Classification	Camera service ELIB function		
Function name	Primary flip control	Symbol	Elib_CAM_Ctrl_Primary_Flip
Function overview			

Flips the specified (primary) surface.

```
                    /* Return value */
#define D_ELIB_CAM_OK                /* Normal end */
#define D_ELIB_CAM_PARAM_NG          /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG  /* Other control action in progress */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Ctrl_Primary_Flip(unsigned int ap_id, HmiSurfID surf, unsigned int number, HmiVmSurfFlipFlags flags);

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
surf	HmiSurfID	In	Flipping target surface ID * See Remarks for detailed information on the settings.
number	unsigned int	in	Buffer number that identifies the front buffer * See Remarks for detailed information on the settings.
flags	HmiVmSurfFlipFlags	in	Flip operation flag * See Remarks for detailed information on the settings.
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG

Remarks

- Before you can call the primary flip control function, you need to acquire camera resources. If the function is called without resource acquirement, it does not flip the primary surface, but returns abnormally with a Resources not acquired error.
- If an attempt is made to control primary surface flipping on a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error.
- If an attempt is made to call the primary flip control function when the overlay setting control function with the primary synchronization flag set to no primary synchronization (HMI_FALSE) is active, this

function returns abnormally with a Handler error.
- When primary surface flip operation is completed, the caller application is notified of an event indicating a primary flip completion notification.
Detailed parameter explanation:
HmiSurfID surf; //The surface ID of the primary surface is specified as the flipping target surface ID.
Primary surface D_HMI_SURF_PRIMARY
unsigned int number; //Front buffer number (fixed at buffer number 1)
To make number effective, D_HMI_VM_SFF_BUFF must be assigned to flags.
Otherwise, the number parameter will be treated as invalid. As a result, the buffer next to the current front buffer will be the front buffer.

HmiVmSurfFlipFlags flags; //Specify the flip-related flags as the bit-wise logical sum (HmiVmSurfFlipFlags).
D_HMI_VM_SFF_NOVSYNC
Front buffer parameter enabling flag D_HMI_VM_SFF_BUFF
* For the specification regarding overlay setting information, see the Video Handler Interface Specifications.
* An example of using the primary flip control function included in the ELIB camera control service is provided below.
Example:
```
int ret; //Return information
HmiSurfID surf; //Flipping target surface ID
unsigned int number; //Front buffer number
HmiVMSurfFlipFlags flags; //Flip setting flags
    surf = PRIMARY_SURF_ID; //Primary surface ID
    number = 1; //Front buffer number (always 1)
    flags = HmiVmSurfFF_buff; //Flip setting flag (Buff enabled)
                    //Primary flip control
    ret = Elib_CAM_Ctrl_Primary_Flip( apID, surf, number, flags );
                    //After calling the function, the caller waits for a primary flip completion notification.
```

1.13 Synchronous overlay reset/set control

Classification	Camera service ELIB function		
Function name	Overlay reset/set control	Symbol	Elib_CAM_Ctrl_Sync_Overlay_Reset
Function overview			
Sets an overlay based on the overlay reset/set information specified as synchronous. #define D_ELIB_CAM_DEV_ID_MAIN_LCD /* Main LCD device ID */ /* Return value */ #define D_ELIB_CAM_OK /* Normal end */ #define D_ELIB_CAM_HANDLER_NG /* Handler error */ #define D_ELIB_CAM_PARAM_NG /* Parameter error */ #define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */ #define D_ELIB_CAM_OTHER_CTL_ACTION_NG /* Other control action in progress */			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Ctrl_Sync_Overlay_Reset(unsigned int ap_id, HmiVmDevID dev, const HmiVmOverlayParam *param)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
dev	HmiVmDevID	In	Overlay-applicable device ID Main LCD: D_ELIB_CAM_DEV_ID_MAIN_LCD

*param	Const HmiVmOverlayParam	In	Overlay setting information structure
			typedef struct HmiVmOverlayParam_t {
			unsigned int surfNum; //Number of resulting surfaces
			HmiVmOverlayParamSurf *surf; // Setting information area for each surface
			HmiBoolean sync; // Synchronous flip flag
			}HmiVmOverlayParam;
			* Details about this function are similar to those of the B-5 overlay setting control function.

Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG

Remarks

- When the overlay reset/set operation is completed, this function returns control to the caller application.,
- If this function (as a synchronous function) is called within an application, it must be called before any asynchronous counterpart is called.
If a synchronous function is called after an asynchronous counterpart, its actual control action will not be performed because control is still in effect when the asynchronous function returns control to the caller and, as a consequence, the call to the synchronous function without confirming a control completion event encounters the Other control action in progress situation during the judgment steps performed by the function itself.
- Before you can call the synchronous overlay reset control function, you need to acquire camera resources. If the function is called without resource acquirement, it does not set an overlay, but returns abnormally with a Resources not acquired error.
- If an attempt is made to set an overlay on a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error.
- The caller application needs to allocate areas for the number of surfaces on which the overlay is to be reset/set. (Areas for up to three surfaces need to be allocated.)
- The surfaces are assigned priorities so that Surf member elements closer to the beginning (first element) are assigned higher priorities.However, graphic surface overlay information always needs to be set in the beginning of element members.
For detailed information on combinations with other surfaces, see the Video Handler Interface Specifications.
- For detailed parameter explanation, see "B-5 Overlay setting control."

1.14 Camera zoom direct control

Classification	Camera control service ELIB function		
Function name	Camera zoom direct control	Symbol	Elib_CAM_Ctrl_Camera_Zoom_D
Function overview			

Controls camera zoom ratio switching.
Unlike Elib_CAM_Ctrl_Camera_Zoom, this function allows you to directly specify absolute values.
* The number of zoom steps varies depending on the camera type and the image resolution.
#define D_ELIB_CAM_TYPE_SIDE /* Back camera */
#define D_ELIB_CAM_TYPE_FRONT /* Front camera */
The specifiable zoom values depend on the recording resolution specified for each camera mode.

Camera / Recording Resolution	Front Camera	Back Camera
UXGA	—	0 ~ 15
SXGA	—	0 ~ 15
VGA	—	0 ~ 15
CIF/Standby	0	0 ~ 15
JAVA format	0	0 ~ 15
QCIF	0 ~ 1	0 ~ 15
Sub-QCIF	0 ~ 1	0 ~ 15

/* Return value */
#define D_ELIB_CAM_OK /* Normal end */
#define D_ELIB_CAM_HANDLER_NG /* Handler error */
#define D_ELIB_CAM_ZOOM_MAX_NG /* Zoom maximum error */
#define D_ELIB_CAM_ZOOM_MIN_NG /* Zoom minimum error */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG /* Camera power off */
#define D_ELIB_CAM_PARAM_NG /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Ctrl_Camera_Zoom_D(unsigned int ap_id, int camera_type, int zoom_val)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	Int	In	Camera type Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE
zoom_val	Int	In	Specifies the number of zoom steps.

			For the range of specifiable values, see Function overview.
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Zoom maximum error : D_ELIB_CAM_ZOOM_MAX_NG Zoom minimum error : D_ELIB_CAM_ZOOM_MIN_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off : D_ELIB_CAM_POWER_OFF_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

Remarks:

- Before you can call the camera zoom direct control function, you need to acquire camera resources. If the function is called without resource acquirement, it does not initiate camera device control, but returns abnormally with a Resources not acquired error.
- If an attempt is made to control zooming of a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error.
- If an attempt is made to control zooming of a camera device whose power is off, this function returns abnormally with a Camera power off error.
- If an attempt is made to perform a zoom control action beyond the maximum number of zoom steps, this function returns abnormally with a Zoom maximum error.
- If an attempt is made to perform a zoom control action below the minimum number of zoom steps, this function returns abnormally with a Zoom minimum error.
- When the zoom control action on each camera device is completed, the caller application is notified of an event indicating a camera zoom control response notification.

Processing:

- Check that the specified number of zoom steps is valid.
- Zoom control action is performed using the [Middle Layer] camera device parameter setting function based on the specified number of zoom steps.
- Camera device parameter setting results are received from the [Middle Layer] VH via a callback function and the zoom management information is updated with the number of zoom steps that is now included in the new camera device configuration.
- The number of current zoom steps and the total number of zoom steps are assigned to the zoom control response event and the caller application is notified of the control result.
- The zoom control response event reports the zoom control action result and the zoom information (the number of current zoom steps and the total number of zoom steps).

Middle layer interface:

- [Middle Layer] camera zoom control response callback function

1.15 Still picture mode setting control

Classification	Camera control service ELIB function		
Function name	Still picture photographing mode setting control	Symbol	Elib_CAM_Ctrl_SetRecordMode
Function overview			
Controls the still picture photographing mode setting. #define D_ELIB_CAM_PHOTO_MODE /* Still picture photographing mode */ #define D_ELIB_CAM_MANUAL_MODE /* Manual burst mode */			
Include File	srv_cam.h		
Calling Sequence	void Elib_CAM_Ctrl_SetRecordMode(UH mode)		

Argument name	Type	I/O	Description
mode	UH	In	Still picture photographing mode D_ELIB_CAM_PHOTO_MODE: Still picture photographing mode D_ELIB_CAM_MANUAL_MODE: Manual burst mode
Return value	Type	I/O	Description
–	-	-	-

Remarks

Remarks:

- This function must be called during the period from the time the camera on/off control (Elib_CAM_Ctrl_Camera_Active) function is called to turn on the camera to the time the photographing-time moving picture control service function is called.

Processing:

- Based on the specified still picture photographing mode, this function performs still picture photographing mode setting control action using the [Middle Layer] still picture photographing mode setting handler.

1.16 Recording resolution setting

Classification	Camera service ELIB function		
Function name	Recording resolution setting	Symbol	Elib_CAM_Set_Record_Resolution
Function overview			

Assigns the recording resolution that applies to the specified camera type (back camera or front camera) and corresponds to each recording mode, to the camera configuration table and to the camera device.

```
#define D_ELIB_CAM_TYPE_SIDE            /* Back camera */
#define D_ELIB_CAM_TYPE_FRONT           /* Front camera */
#define D_ELIB_CAM_SNAP                 /* Still picture */
#define D_ELIB_CAM_MOVIE                /* Moving picture */
#define D_ELIB_CAM_BURST_SNAP           /* Burst shooting */
#define D_ELIB_CAM_RESOLUTION_VGA       /* VGA format */
#define D_ELIB_CAM_RESOLUTION_CIF       /* CIF format */
#define D_ELIB_CAM_RESOLUTION_QVGA      /* QVGA format (not used) */
#define D_ELIB_CAM_RESOLUTION_QCIF      /* QCIF format */
#define D_ELIB_CAM_RESOLUTION_QQVGA     /* QQVGA format (not used) */
#define D_ELIB_CAM_RESOLUTION_SUB_QCIF  /* sub_QCIF format */
#define D_ELIB_CAM_RESOLUTION_WAITSCREEN  /* Standby format */
#define D_ELIB_CAM_RESOLUTION_JAVA /* JAVA format */
#define D_ELIB_CAM_RESOLUTION_UXGA      /* UXGA format */
#define D_ELIB_CAM_RESOLUTION_SXGA      /* SXGA format */
                        /* Return value */
#define D_ELIB_CAM_OK                   /* Normal end */
#define D_ELIB_CAM_HANDLER_NG           /* Handler error */
#define D_ELIB_CAM_DRIVER_NG            /* Driver error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG  /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG         /* Camera power off */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Set_Record_Resolution(unsigned int ap_id, int camera_type, int record_mode,int resolution)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	Int	In	Camera type Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE

record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE Burst: D_ELIB_CAM_BURST_SNAP
resolution	Int	In	Recording resolution VGA :D_ELIB_CAM_RESOLUTION_VGA CIF :D_ELIB_CAM_RESOLUTION_CIF QVGA : D_ELIB_CAM_RESOLUTION_QVGA (not used) QCIF : D_ELIB_CAM_RESOLUTION_QCIF QQVGA :D_ELIB_CAM_RESOLUTION_QQVGA (not used) Sub-QCIF:D_ELIB_CAM_RESOLUTION_SUB_QCIF Standby: D_ELIB_CAM_RESOLUTION_WAITSCREEN JAVA: D_ELIB_CAM_RESOLUTION_JAVA UXGA: D_ELIB_CAM_RESOLUTION_UXGA SXGA : D_ELIB_CAM_RESOLUTION_SXGA
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Handler error : D_ELIB_CAM_HANDLER_NG Driver error : D_ELIB_CAM_DRIVER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress (the function only stores the setting in the camera configuration table) : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off (the function only stores the setting in the camera configuration table). : D_ELIB_CAM_POWER_OFF_NG

Remarks

- See 1.16 for the initial value.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If an attempt is made to set a recording resolution on a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error. However, the setting is stored in the camera configuration table.
- If an attempt is made to set a recording resolution on a camera device whose power is off, this function returns abnormally with a Camera power off error. However, the setting is stored in the camera configuration table.
- The setting is stored in the area corresponding to the pertinent camera type and recording mode, of the camera configuration table.
- The specified camera device is notified that it now contains recording resolution data.
- When the recording resolution setting is updated on a camera device, zoom reset control action is also performed on that camera device at the same time.
Also, the settings in the camera mode dependent default zoom value table are reset.

- When a recording resolution is set on a camera device, the caller application is notified of an event indicating a recording resolution setting completion notification.
- Only when the function is called from a camera application, the function stores the setting not only in the camera configuration table as explained above, but also in non-volatile memory.
However, if the recording resolution is JAVA (D_ELIB_CAM_RESOLUTION_JAVA), the function only stores the data in the camera configuration table, and does not store it in non-volatile memory.
(Settings stored in non-volatile memory will be used as initial values if the caller application is a camera application.))

1.17 Brightness setting

<table>
<tr><td>Classification</td><td colspan="4">Camera service ELIB function</td></tr>
<tr><td>Function name</td><td>Brightness setting</td><td>Symbol</td><td colspan="2">Elib_CAM_Set_Brightness</td></tr>
<tr><td>Function overview</td><td colspan="4"></td></tr>
<tr><td colspan="5">Assigns brightness settings to the camera device and stores them in the camera configuration.

#define D_ELIB_CAM_BRIGHTNESS_LEVEL1 /* Brightness level 1 (-2) */
#define D_ELIB_CAM_BRIGHTNESS_LEVEL2 /* Brightness level 2 (-1) */
#define D_ELIB_CAM_BRIGHTNESS_LEVEL3 /* Brightness level 3 (0) */
#define D_ELIB_CAM_BRIGHTNESS_LEVEL4 /* Brightness level 4 (+1) */
#define D_ELIB_CAM_BRIGHTNESS_LEVEL5 /* Brightness level 5 (+2) */
/* Return value */
#define D_ELIB_CAM_OK /* Normal end */
#define D_ELIB_CAM_HANDLER_NG /* Handler error */
#define D_ELIB_CAM_DRIVER_NG /* Driver error */
#define D_ELIB_CAM_PARAM_NG /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG /* Camera power off */</td></tr>
<tr><td>Include File</td><td colspan="4">srv_cam.h</td></tr>
<tr><td>Calling Sequence</td><td colspan="4">int Elib_CAM_Set_Brightness(unsigned int ap_id, int level)</td></tr>
<tr><td>Argument name</td><td>Type</td><td>I/O</td><td colspan="2">Description</td></tr>
<tr><td>ap_id</td><td>unsigned int</td><td>In</td><td colspan="2">Application ID</td></tr>
<tr><td>level</td><td>Int</td><td>In</td><td colspan="2">Brightness level
Level 1 (-2): D_ELIB_CAM_BRIGHTNESS_LEVEL1
Level 2 (-1): D_ELIB_CAM_BRIGHTNESS_LEVEL2
Level 3 (0): D_ELIB_CAM_BRIGHTNESS_LEVEL3
Level 4 (+1): D_ELIB_CAM_BRIGHTNESS_LEVEL4
Level 5 (+2): D_ELIB_CAM_BRIGHTNESS_LEVEL5</td></tr>
<tr><td>Return value</td><td>Type</td><td>I/O</td><td colspan="2">Description</td></tr>
<tr><td>Ret</td><td>Int</td><td>Out</td><td colspan="2">Normal end : D_ELIB_CAM_OK
Handler error: D_ELIB_CAM_HANDLER_NG
Driver error : D_ELIB_CAM_DRIVER_NG
Parameter error : D_ELIB_CAM_PARAM_NG
Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG
Other control action in progress (the function only stores the setting in the camera</td></tr>
</table>

			configuration table) : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off (the function only stores the setting in the camera configuration table). : D_ELIB_CAM_POWER_OFF_NG

Remarks

- The initial value is set at the brightness level 3 (0) (see 1.16).
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If an attempt is made to set a brightness level on a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error. However, the setting is stored in the camera configuration table.
- If an attempt is made to set a brightness level on a camera device whose power is off, this function returns abnormally with a Camera power off error. However, the setting is stored in the camera configuration table.
- The setting data will be retained as long as the camera resources are allocated and will be initialized when the camera resources are freed.
- The specified camera device is notified that it now contains brightness level data.
- If you only want to assign brightness data to the camera device, use the B-3 camera image adjustment view control function.
- When a brightness level is set on a camera device, the caller application is notified of an event indicating a brightness setting completion notification.
- Only when the function is called from a video phone application, the function stores the setting not only in the camera configuration table as explained above, but also in non-volatile memory.
(Settings stored in non-volatile memory will be used as initial values only if the caller application is a video phone application.))

1.18 Color tone setting

Classification	Camera service ELIB function		
Function name	Color tone setting	Symbol	Elib_CAM_Set_Color_Tone
Function overview			

Assigns color tone data to the camera device and stores tit in in the camera configuration.

```
#define D_ELIB_CAM_COLOR_TONE_SEPIA       /* Sepia */
#define D_ELIB_CAM_COLOR_TONE_NORMAL      /* Ordinary */
#define D_ELIB_CAM_COLOR_TONE_MONOCHROME  /* Black and white */
                                  /* Return value */
#define D_ELIB_CAM_OK                     /* Normal end */
#define D_ELIB_CAM_HANDLER_NG             /* Handler error */
#define D_ELIB_CAM_DRIVER_NG              /* Driver error */
#define D_ELIB_CAM_PARAM_NG               /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG     /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG    /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG           /* Camera power off */
```

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Set_Color_Tone(unsigned int ap_id, int tone_type)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
tone_type	Int	In	Color tone type Sepia : D_ELIB_CAM_COLOR_TONE_SEPIA Ordinary : D_ELIB_CAM_COLOR_TONE_NORMAL Black and white : D_ELIB_CAM_COLOR_TONE_MONOCHROME
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Driver error : D_ELIB_CAM_DRIVER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress (the function only stores the setting in the camera configuration table) : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off (the function only stores the setting in the

			camera configuration table). : D_ELIB_CAM_POWER_OFF_NG

Remarks

- The initial value is set at Ordinary (see 1.16).
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If an attempt is made to set a color tone a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error. However, the setting is stored in the camera configuration table.
- If an attempt is made to set a color tone on a camera device whose power is off, this function returns abnormally with a Camera power off error. However, the setting is stored in the camera configuration table.
- The setting data will be retained as long as the camera resources are allocated and will be initialized when the camera resources are freed.
- The specified camera device is notified that it now contains color tone data.
- If you only want to assign color tone data to the camera device, use the B-6 camera image adjustment view control function.
- When a color tone is set on a camera device, the caller application is notified of an event indicating a color tone setting completion notification.

1.19 White balance adjustment setting

Classification	Camera service ELIB function		
Function name	Camera white balance adjustment setting	Symbol	Elib_CAM_Set_White_Balance
Function overview			

Assigns white balance adjustment data to the camera device and stores it in the camera configuration.

```
#define D_ELIB_CAM_WB_FINE_WEATHER        /* Fine */
#define D_ELIB_CAM_WB_CLOUDY_WEATHER      /* Cloud */
#define D_ELIB_CAM_WB_AUTO                /* Auto */
#define D_ELIB_CAM_WB_ELECTRIC_LIGHT      /* Bulb */
                                  /* Return value */
#define D_ELIB_CAM_OK                     /* Normal end */
#define D_ELIB_CAM_HANDLER_NG             /* Handler error */
#define D_ELIB_CAM_DRIVER_NG              /* Driver error */
#define D_ELIB_CAM_PARAM_NG               /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG     /* Resource not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG    /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG           /* Camera power off */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Set_White_Balance(unsigned int ap_id, int wb)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
wb	Int	In	White balance adjustment flag Fine : D_ELIB_CAM_WB_FINE_WEATHER Cloud : D_ELIB_CAM_WB_CLOUDY_WEATHER Auto: D_ELIB_CAM_WB_AUTO Bulb : D_ELIB_CAM_WB_ELECTRIC_LIGHT
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Handler error : D_ELIB_CAM_HANDLER_NG Driver error : D_ELIB_CAM_DRIVER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress (the function only stores the setting in the camera configuration table) :

			D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off (the function only stores the setting in the camera configuration table). : D_ELIB_CAM_POWER_OFF_NG

Remarks

- The initial value is set at Auto (see 1.16).
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If an attempt is made to set a white balance on a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error. However, the setting is stored in the camera configuration table.
- If an attempt is made to set a white balance on a camera device whose power is off, this function returns abnormally with a Camera power off error. However, the setting is stored in the camera configuration table.
- The setting is stored in the camera configuration data.
- The specified camera device is notified that it now contains white balance data.
- If you only want to assign white balance data to the camera device, use the B-3 camera image adjustment view control function.
- When a white balance is set on a camera device, the caller application is notified of an event indicating a white balance data setting completion notification.
- Only when the function is called from a camera application or video phone application, the function stores the setting not only in the camera configuration table as explained above, but also in non-volatile memory.

(Settings stored in non-volatile memory will be used as initial values only if the caller application is a camera application or video phone application.)
(Settings stored in non-volatile memory are common to camera and video phone applications.)

1.20 Photographing mode setting

Classification	Camera service ELIB function		
Function name	Photographing mode setting	Symbol	Elib_CAM_Set_Photo_Mode
Function overview			

Assigns a photographing mode for each camera type to the camera device and stores it in the camera configuration.

```
#define D_ELIB_CAM_TYPE_FRONT          /* Front camera */
#define D_ELIB_CAM_TYPE_SIDE           /* Back camera */
#define D_ELIB_CAM_MODE_CONTACT        /* Burst mode */
#define D_ELIB_CAM_MODE_PORTRAIT       /* Portrait mode */
#define D_ELIB_CAM_MODE_SCENERY        /* Scenery mode */
#define D_ELIB_CAM_MODE_NORMAL         /* Standard mode */
#define D_ELIB_CAM_MODE_NIGHT          /* Night mode */
#define D_ELIB_CAM_MODE_OCR            /* OCR mode */
#define D_ELIB_CAM_MODE_BARCODE        /* Two-dimensional bar code mode */
#define D_ELIB_CAM_MODE_SPORTS         /* Sports mode */
                        /* Return value */
#define D_ELIB_CAM_OK                  /* Normal end */
#define D_ELIB_CAM_HANDLER_NG          /* Handler error */
#define D_ELIB_CAM_DRIVER_NG           /* Driver error */
#define D_ELIB_CAM_PARAM_NG            /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG  /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG        /* Camera power off */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Set_Photo_Mode(unsigned int ap_id, int camera_type, int photo_mode)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	Int	In	Camera type Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE
photo_mode	Int	In	Photographing mode Close-up : D_ELIB_CAM_MODE_CONTACT Portrait : D_ELIB_CAM_MODE_PORTRAIT Scenery : D_ELIB_CAM_MODE_SCENERY (Standard: D_ELIB_CAM_MODE_NORMAL) * Equivalent to Scenery (to match text in the operation

			specifications) Night mode : D_ELIB_CAM_MODE_NIGHT OCR : D_ELIB_CAM_MODE_OCR Two-dimensional bar code: D_ELIB_CAM_MODE_BARCODE Sports mode: D_ELIB_CAM_MODE_SPORTS
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Driver error : D_ELIB_CAM_DRIVER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resource not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress (the function only stores the setting in the camera configuration table) : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off (the function only stores the setting in the camera configuration table). : D_ELIB_CAM_POWER_OFF_NG

Remarks

- The initial value for the front camera is set at Portrait. The initial value for the back camera is set at Scenery (Standard).
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If an attempt is made to set a photographing mode a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error. However, the setting is stored in the camera configuration table.
- If an attempt is made to set a photographing mode a camera device whose power is off, this function returns abnormally with a Camera power off error. However, the setting is stored in the camera configuration table.
- The setting data will be retained as long as the camera resources are allocated and will be initialized when the camera resources are freed.
- The specified camera device is notified that it now contains photographing mode data.
- When a photographing mode is set on a camera device, the caller application is notified of an event indicating a photographing mode data setting completion notification.

1.21 Burst shot interval setting

<table>
<tr><td>Classification</td><td colspan="4">Camera service ELIB function</td></tr>
<tr><td>Function name</td><td>Burst shot interval setting</td><td>Symbol</td><td colspan="2">Elib_CAM_Set_Burst_Speed</td></tr>
<tr><td>Function overview</td><td colspan="4"></td></tr>
<tr><td colspan="5">If the recording style is Burst, this function assigns a burst shot interval to the camera configuration table and stores it non-volatile memory and in the camera device.
#define D_ELIB_CAM_BURST_SPEED_SLOW /* 2.0 seconds */
#define D_ELIB_CAM_BURST_SPEED_QUICK /* 1.0 second */
#define D_ELIB_CAM_BURST_SPEED_MACH /* 0.5 second */
/* Return value */
#define D_ELIB_CAM_OK /* Normal end */
#define D_ELIB_CAM_HANDLER_NG /* Handler error */
#define D_ELIB_CAM_DRIVER_NG /* Driver error */
#define D_ELIB_CAM_PARAM_NG /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG /* Camera power off */</td></tr>
<tr><td>Include File</td><td colspan="4">srv_cam.h</td></tr>
<tr><td>Calling Sequence</td><td colspan="4">int Elib_CAM_Set_Burst_Speed(unsigned int ap_id, int mode)</td></tr>
<tr><td>Argument name</td><td>Type</td><td>I/O</td><td colspan="2">Description</td></tr>
<tr><td>ap_id</td><td>unsigned int</td><td>In</td><td colspan="2">Application ID</td></tr>
<tr><td>mode</td><td>Int</td><td>In</td><td colspan="2">Burst mode
2.0 seconds: D_ELIB_CAM_BURST_SPEED_SLOW
1.0 second: D_ELIB_CAM_BURST_SPEED_QUICK
0.5 second: D_ELIB_CAM_BURST_SPEED_MACH</td></tr>
<tr><td>Return value</td><td>Type</td><td>I/O</td><td colspan="2">Description</td></tr>
<tr><td>Ret</td><td>Int</td><td>Out</td><td colspan="2">Normal end : D_ELIB_CAM_OK
Handler error: D_ELIB_CAM_HANDLER_NG
Driver error : D_ELIB_CAM_DRIVER_NG
Parameter error : D_ELIB_CAM_PARAM_NG
Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG
Other control action in progress (the function only stores the setting in non-volatile memory) : D_ELIB_CAM_OTHER_CTL_ACTION_NG
Camera power off (the function only stores the setting in non-volatile memory) : D_ELIB_CAM_POWER_OFF_NG</td></tr>
</table>

Remarks

- See 1.16 for the initial value that will be effective when non-volatile memory is reset.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If an attempt is made to set a burst shot interval on a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error. However, the setting is stored in non-volatile memory.
- If an attempt is made to set a burst shot interval on a camera device whose power is off, this function returns abnormally with a Camera power off error. However, the setting is stored in non-volatile memory.
- The setting data is stored in non-volatile memory.
- The specified camera device is notified that it now contains burst shot interval data.
- If you only want to assign burst shot interval data to the camera device, use the B-3 camera image adjustment view control function.
- When a burst shot interval is set on a camera device, the caller application is notified of an event indicating a burst shot interval setting completion notification.

1.22 Camera off timer setting (not used)

Classification	Camera service ELIB function		
Function name	Camera off timer setting	Symbol	Elib_CAM_Set_OffTimer
Function overview			

Stores camera auto off timer information in non-volatile memory.

```
#define D_ELIB_CAM_OFF_TIMER_ON          /* Camera off timer enabled */
#define D_ELIB_CAM_OFF_TIMER_OFF         /* Camera off timer disabled */
                                 /* Return value */
#define D_ELIB_CAM_OK                    /* Normal end */
#define D_ELIB_CAM_DRIVER_NG             /* Driver error */
#define D_ELIB_CAM_PARAM_NG              /* Parameter error */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Set_OffTimer(unsigned int ap_id, T_ELIB_CAM_OFF_TIMER *timer)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
*timer	T_ELIB_CAM_OFF_TIMER	In	Pointer to the camera off time setting data structure typedef struct t_elib_cam_off_timer { int on_off; /* Off timer setting Timer enabled : D_ELIB_CAM_OFF_TIMER_ON Timer disabled : D_ELIB_CAM_OFF_TIMER_OFF int timer; /* Timer value (1 to 60 minutes) */ } T_ELIB_CAM_OFF_TIMER; * If the timer is disabled, the specified timer value will be treated as invalid.
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG

Remarks

- The initial settings are timer enabled and a timer value of 5 minutes.
- The setting data is stored in non-volatile memory.
- Since it is assumed that the camera off timer is set on the overall configuration screen, the off timer can be used regardless of whether camera resources are acquired.

1.23 Record size setting

Classification	Camera service ELIB function		
Function name	Record size setting	Symbol	Elib_CAM_Set_Record_Size

Function overview

Stores record size information in non-volatile memory.

```
#define D_ELIB_CAM_REC_SIZE_MAIL_ATTACH    /* Moving picture: Mail */
#define D_ELIB_CAM_REC_SIZE_NO_LIMIT       /* Moving picture: Moving picture memo   */
#define D_ELIB_CAM_REC_SIZE_LONG_TIME /* Moving picture: Long time */
                                   /* Return value */
#define D_ELIB_CAM_OK                      /* Normal end */
#define D_ELIB_CAM_DRIVER_NG               /* Driver error */
#define D_ELIB_CAM_PARAM_NG                /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG      /* Resources not acquired */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Set_Record_Size(unsigned int ap_id, int rec_size)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
rec_size	int	In	Record size ・Moving picture record size Mail : D_ELIB_CAM_REC_SIZE_MAIL_ATTACH Moving picture memo : D_ELIB_CAM_REC_SIZE_NO_LIMIT Long time: D_ELIB_CAM_REC_SIZE_LONG_TIME

Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- See 1.16 for the initial value that will be effective when non-volatile memory is reset.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- The setting data is stored in non-volatile memory.

1.24 Recording medium setting

Classification	Camera service ELIB function		
Function name	Camera recording medium setting	Symbol	Elib_CAM_Set_Record_Media
Function overview			

Assigns recording mode dependent recording medium information to the camera configuration table.

```
#define D_ELIB_CAM_SNAP                 /* Still picture */
#define D_ELIB_CAM_MOVIE                /* Moving picture */
#define D_ELIB_CAM_RECORD_INSIDE        /* Internal memory */
                                /* Return value */
#define D_ELIB_CAM_OK                   /* Normal end */
#define D_ELIB_CAM_DRIVER_NG            /* Driver error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
```

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Set_Record_Media(unsigned int ap_id, int recoerd_mode, int media)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE
media	int	In	Recording medium flag Internal memory: D_ELIB_CAM_RECORD_INSIDE
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The setting data will be retained as long as the camera resources are allocated and will be initialized when the camera resources are freed.
- The initial value is set at Internal memory.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.25 Recording quality setting

Classification	Camera service ELIB function		
Function name	Recording quality setting	Symbol	Elib_CAM_Set_Record_Quality
Function overview			
Stores an image recording quality flag that is pertinent to the recording mode (still or moving) and the recording media (inside/SD), in non-volatile memory. #define D_ELIB_CAM_SNAP /* Still picture */ #define D_ELIB_CAM_MOVIE /* Moving picture */ #define D_ELIB_CAM_RECORD_INSIDE /* Internal memory */ #define D_ELIB_CAM_QUALITY_ECONOMY /* Still picture: Mail restriction; Moving picture: Priority on time */ #define D_ELIB_CAM_QUALITY_NORMAL /* Still picture: Large-size mail restriction; Moving picture: Standard */ #define D_ELIB_CAM_QUALITY_FINE /* Still picture: No restriction; Moving picture: Priority on picture quality */ /* Return value */ #define D_ELIB_CAM_OK /* Normal end */ #define D_ELIB_CAM_DRIVER_NG /* Driver error */ #define D_ELIB_CAM_PARAM_NG /* Parameter error */ #define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Set_Record_Quality(unsigned int ap_id, int record_mode, int media, int quality)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE
Media	int	In	Recording medium flag Internal memory: D_ELIB_CAM_RECORD_INSIDE * Linux PF allows only Internal memory to be specified.
Quality	int	In	Recording quality flag Mail restriction (still pictures); Priority on time (moving pictures) : D_ELIB_CAM_QUALITY_ECONOMY Large-size mail restriction (still pictures); Standard (moving pictures) : D_ELIB_CAM_QUALITY_NORMAL No restriction (still pictures); Priority on picture quality (moving pictures)

			: D_ELIB_CAM_QUALITY_FINE
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG
Remarks			

- See 1.16 for the initial value that will be effective when non-volatile memory is reset.
- The setting data is stored in the area corresponding to the recording mode and the recording medium flag of non-volatile memory. However, if the recording mode is Still, the recording medium flag is invalid.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- In Still picture mode, the Mail restriction, Large-size mail restriction, and No restriction options are valid for the recording quality flag.
- In Moving picture mode, the Priority on time, Standard, and Priority on picture quality options are valid for the recording quality flag.

1.26 Saving of various camera settings

Classification	Camera service ELIB function		
Function name	Saving of various camera settings	Symbol	Elib_CAM_Set_General_Conf
Function overview			

Saves various general settings to be used for camera applications in non-volatile memory.
The following settings can be saved:

```
typedef struct tagELIB_CAM_GENERALCONF {
    BYTE    bSelfTimer      Self-timer duration in seconds
    BYTE    bShutSound_ST   Still picture/burst shot shutter sound
    BYTE    bShutSound_MV   Moving picture shutter sound
    BYTE    bBurstMode      Burst shooting type
  BYTE    bAutoSave       Automatic registration setting
BYTE    bDispSize: View size
} _ELIB_CAM_GENERALCONF;
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Set_General_Conf (unsigned int ap_id, _ELIB_CAM_GENERALCONF Conf);

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Conf	_ELIB_CAM_GENERALCONF	In	Information structure for various camera settings

Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- See 1.16 for the initial value that will be effective when non-volatile memory is reset.
- The setting data is stored in non-volatile memory.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.27 Default startup camera settings

Classification	Camera service ELIB function		
Function name	Default start-up camera settings	Symbol	Elib_CAM_Set_Def_Camera
Function overview			
Stores the default starting camera setting in non-volatile memory.			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Set_Def_Camera (unsigned int ap_id , int type);		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Type	int	In	Default starting camera Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG
Remarks			

- The setting data is stored in non-volatile memory.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.28 Camera mode dependent default zoom value setting

Classification	Camera service ELIB function		
Function name	Camera mode dependent default zoom value setting	Symbol	Elib_CAM_Set_Def_Zoom
Function overview			

Sets the initial zoom value that corresponds to the camera type (front or back) and the recording mode (Still, Moving, or Burst).
The value set by this function is stored in the camera mode dependent default zoom value table.
The camera mode dependent default zoom value table is initialized to no-zoom (0) when the camera resources are acquired.
The specifiable zoom values depend on the recording resolution specified for each camera mode.

Camera / Recording Resolution	Front Camera	Back Camera
UXGA	—	0 ~ 15
SXGA	—	0 ~ 15
VGA	—	0 ~ 15
CIF/Standby	0	0 ~ 15
JAVA format	0	0 ~ 15
QCIF	0 ~ 1	0 ~ 15
Sub-QCIF	0 ~ 1	0 ~ 15

If the recording resolution is changed by a call to Elib_CAM_Set_Record_Resolution or Elib_CAM_SetRecord_Resolution_All, the zoom value corresponding to that camera mode is initialized to no-zoom (0).
The camera mode dependent default zoom value table is read when the power is turned on by Elib_CAM_Ctrl_Camera_Active or when the recording mode is changed by Elib_CAM_Ctrl_Record_Mode_Change.
The camera device will be initialized with the camera mode dependent zoom value that is stored in this table.

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Set_Def_Zoom (unsigned int ap_id, int camera_type, int record_mode, int zoom);		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	int	In	Camera type Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE

record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE Burst: D_ELIB_CAM_BURST_SNAP
Zoom	int	In	Number of zoom steps * See Function overview for information on the range of specifiable values.
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG
Remarks			
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.			

1.29 Burst shot count setting

Classification	Camera Elib function		
Function name	Burst shot count setting	Symbol	Elib_CAM_Set_SeqShoot_Numbers

Function overview

Assigns the number of burst shots (4 to 20) to the camera device and stores it in non-volatile memory.

```
/* Return value */
#define D_ELIB_CAM_OK                   /* Normal end */
#define D_ELIB_CAM_HANDLER_NG           /* Handler error */
#define D_ELIB_CAM_DRIVER_NG            /* Driver error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG  /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG         /* Camera power off */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Set_SeqShoot_Numbers(unsigned int Ap_id, int Numbers)

Argument name	Type	I/O	Description
Ap_id	unsigned int	In	Application ID
Numbers	int	In	Number of burst shots (D_ELIB_CAM_SEQSHTNUM_MIN to D_ELIB_CAM_SEQSHTNUM_MAX)
Return value	**Type**	**I/O**	**Description**
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Driver error : D_ELIB_CAM_DRIVER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress (the function only stores the setting in non-volatile memory) : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off (the function only stores the setting in non-volatile memory) : D_ELIB_CAM_POWER_OFF_NG

Remarks

- The initial value is set at 5.
- The setting data is stored in non-volatile memory and the camera device.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

- If an attempt is made to set a burst shot count on a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error. However, the setting is stored in non-volatile memory.
- If an attempt is made to set a burst shot count on a camera device whose power is off, this function returns abnormally with a Camera power off error. However, the setting is stored in non-volatile memory.
- The setting data is stored in non-volatile memory.
- The specified camera device is notified that it now contains burst shot count data.
- When a burst shot count is set on a camera device, the caller application is notified of an event indicating a burst shot count setting completion notification.

1.30 Flicker suppression setting

Classification	Camera Elib function		
Function name	Flicker suppression setting	Symbol	Elib_CAM_Set_Flicker
Function overview			

Assigns the flicker suppression setting to the camera device and stores it in non-volatile memory.

```
/* Return value */
#define D_ELIB_CAM_OK                   /* Normal end */
#define D_ELIB_CAM_HANDLER_NG           /* Handler error */
#define D_ELIB_CAM_DRIVER_NG            /* Driver error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
#define D_ELIB_CAM_OTHER_CTL_ACTION_NG  /* Other control action in progress */
#define D_ELIB_CAM_POWER_OFF_NG         /* Camera power off */
```

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Set_Flicker (unsigned int ap_id, int flicker_mode)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
flicker_mode	int	In	Flicker suppression setting Auto: D_ELIB_CAM_FLICKER_AUTO 50 Hz: D_ELIB_CAM_FLICKER_50 60 Hz: D_ELIB_CAM_FLICKER_60
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Driver error : D_ELIB_CAM_DRIVER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Other control action in progress (the function only stores the setting in non-volatile memory) : D_ELIB_CAM_OTHER_CTL_ACTION_NG Camera power off (the function only stores the setting in non-volatile memory) : D_ELIB_CAM_POWER_OFF_NG

Remarks

- The initial value is set at Auto.

- The flicker suppression setting data is stored in non-volatile memory and the camera device.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If an attempt is made to assign a flicker suppression setting to a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error. However, the setting is stored in non-volatile memory.
- If an attempt is made to assign a flicker suppression setting to a camera device whose power is off, this function returns abnormally with a Camera power off error. However, the setting is stored in non-volatile memory.
- The setting data is stored in non-volatile memory.
- The specified camera device is notified that it now contains flicker suppression setting data.
- When a flicker suppression setting is assigned to a camera device, the caller application is notified of an event indicating a flicker suppression setting completion notification.

1.31 Recording resolution setting status read

Classification	Camera service ELIB function		
Function name	Recording resolution setting status read	Symbol	Elib_CAM_Get_Record_Resolution
Function overview			

Reads the recording resolution that applies to the specified camera type (back camera or front camera) and corresponds to each recording mode, from the camera configuration table.

```
#define D_ELIB_CAM_TYPE_SIDE            /* Back camera */
#define D_ELIB_CAM_TYPE_FRONT           /* Front camera */
#define D_ELIB_CAM_SNAP                 /* Still picture */
#define D_ELIB_CAM_MOVIE                /* Moving picture */
#define D_ELIB_CAM_BURST_SNAP           /* Burst shooting */
#define D_ELIB_CAM_RESOLUTION_VGA       /* VGA format */
#define D_ELIB_CAM_RESOLUTION_CIF       /* CIF format */
#define D_ELIB_CAM_RESOLUTION_QVGA      /* QVGA format (not used) */
#define D_ELIB_CAM_RESOLUTION_QCIF      /* QCIF format */
#define D_ELIB_CAM_RESOLUTION_QQVGA     /* QQVGA format (not used) */
#define D_ELIB_CAM_RESOLUTION_SUB_QCIF  /* sub_QCIF format */
#define D_ELIB_CAM_RESOLUTION_WAITSCREEN  /* Standby format */
#define D_ELIB_CAM_RESOLUTION_JAVA /* JAVA format */
#define D_ELIB_CAM_RESOLUTION_UXGA      /* UXGA format */
#define D_ELIB_CAM_RESOLUTION_SXGA      /* SXGA format */
                        /* Return value */
#define D_ELIB_CAM_DRIVER_NG            /* Driver error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Get_Record_Resolution(unsigned int ap_id, int camera_type, int reocrd_mode)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	int	In	Camera type Back camera: D_ELIB_CAM_TYPE_SIDE Front camera: D_ELIB_CAM_TYPE_FRONT
record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE Burst: D_ELIB_CAM_BURST_SNAP

Return value	Type	I/O	Description
Ret	int	Out	Recording resolution VGA : D_ELIB_CAM_RESOLUTION_VGA CIF : D_ELIB_CAM_RESOLUTION_CIF QVGA : D_ELIB_CAM_RESOLUTION_QVGA (not used) QCIF : D_ELIB_CAM_RESOLUTION_QCIF QQVGA : D_ELIB_CAM_RESOLUTION_QQVGA (not used) Sub_QCIF : D_ELIB_CAM_RESOLUTION_SUB_QCIF Standby : D_ELIB_CAM_RESOLUTION_WAITSCREEN JAVA: D_ELIB_CAM_RESOLUTION_JAVA UXGA :D_ELIB_CAM_RESOLUTION_UXGA SXGA : D_ELIB_CAM_RESOLUTION_SXGA Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The data to be read is stored in the area corresponding to the pertinent camera type and recording mode, of the camera configuration table.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.32 Brightness setting status read

<table>
<tr><td>Classification</td><td colspan="3">Camera service ELIB function</td></tr>
<tr><td>Function name</td><td>Brightness setting status read</td><td>Symbol</td><td>Elib_CAM_Get_Brightness</td></tr>
<tr><td>Function overview</td><td colspan="3"></td></tr>
<tr><td colspan="4">Reads the brightness setting from the camera configuration table.

#define D_ELIB_CAM_BRIGHTNESS_LEVEL1 /* Brightness level 1 (-2) */
#define D_ELIB_CAM_BRIGHTNESS_LEVEL2 /* Brightness level 2 (-1) */
#define D_ELIB_CAM_BRIGHTNESS_LEVEL3 /* Brightness level 3 (0) */
#define D_ELIB_CAM_BRIGHTNESS_LEVEL4 /* Brightness level 4 (+1) */
#define D_ELIB_CAM_BRIGHTNESS_LEVEL5 /* Brightness level 5 (+2) */
/* Return value */
#define D_ELIB_CAM_DRIVER_NG /* Driver error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */</td></tr>
<tr><td>Include File</td><td colspan="3">srv_cam.h</td></tr>
<tr><td>Calling Sequence</td><td colspan="3">int Elib_CAM_Get_Brightness (unsigned int ap_id)</td></tr>
<tr><td>Argument name</td><td>Type</td><td>I/O</td><td>Description</td></tr>
<tr><td>ap_id</td><td>unsigned int</td><td>In</td><td>Application ID</td></tr>
<tr><td>Return value</td><td>Type</td><td>I/O</td><td>Description</td></tr>
<tr><td>Ret</td><td>Int</td><td>Out</td><td>Brightness data
Brightness level
Level 1 (-2): D_ELIB_CAM_BRIGHTNESS_LEVEL1
Level 2 (-1): D_ELIB_CAM_BRIGHTNESS_LEVEL2
Level 3 (0): D_ELIB_CAM_BRIGHTNESS_LEVEL3
Level 4 (+1): D_ELIB_CAM_BRIGHTNESS_LEVEL4
Level 5 (+2): D_ELIB_CAM_BRIGHTNESS_LEVEL5
Driver error : D_ELIB_CAM_DRIVER_NG
Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG</td></tr>
<tr><td>Remarks</td><td colspan="3"></td></tr>
<tr><td colspan="4">- The data to be read is stored in the camera configuration table.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.</td></tr>
</table>

1.33 Color tone setting status read

Classification	Camera service ELIB function		
Function name	Color tone setting status read	Symbol	Elib_CAM_Get_Color_Tone
Function overview			

Reads the color tone setting from the camera configuration table.

```
#define D_ELIB_CAM_COLOR_TONE_SEPIA       /* Sepia */
#define D_ELIB_CAM_COLOR_TONE_NORMAL      /* Ordinary */
#define D_ELIB_CAM_COLOR_TONE_MONOCHROME  /* Black and white */
                                  /* Return value */
#define D_ELIB_CAM_DRIVER_NG              /* Driver error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG     /* Resources not acquired */
```

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Get_Color_Tone(unsigned int ap_id)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	Type	I/O	Description
Ret	Int	Out	Color tone type Sepia : D_ELIB_CAM_COLOR_TONE_SEPIA Ordinary : D_ELIB_CAM_COLOR_TONE_NORMAL Black and white : D_ELIB_CAM_COLOR_TONE_MONOCHROME Driver error : D_ELIB_CAM_DRIVER_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The data to be read is stored in the camera configuration table.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.34 White balance adjustment setting status read

Classification	Camera service ELIB function		
Function name	Camera white balance adjustment setting status read	Symbol	Elib_CAM_Get_White_Balance
Function overview			

Reads the white balance adjustment setting from the camera configuration table.

```
#define D_ELIB_CAM_WB_FINE_WEATHER       /* Fine */
#define D_ELIB_CAM_WB_CLOUDY_WEATHER     /* Cloud */
#define D_ELIB_CAM_WB_AUTO               /* Auto */#define
D_ELIB_CAM_WB_ELECTRIC_LIGHT     /* Bulb */
                                 /* Return value */
#define D_ELIB_CAM_DRIVER_NG             /* Driver error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG    /* Resources not acquired */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Get_White_Balance(unsigned int ap_id)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	Type	I/O	Description
Ret	int	Out	White balance adjustment flag Fine : D_ELIB_CAM_WB_FINE_WEATHER Cloud : D_ELIB_CAM_WB_CLOUDY_WEATHER Auto : D_ELIB_CAM_WB_AUTO Bulb : D_ELIB_CAM_WB_ELECTRIC_LIGHT Driver error : D_ELIB_CAM_DRIVER_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The data to be read is stored in the camera configuration table.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.35 Photographing mode setting status read

Classification	Camera service ELIB function		
Function name	Photographing mode setting status read	Symbol	Elib_CAM_Get_Photo_Mode
Function overview			

Reads the photographing mode setting corresponding to the specified camera type from the camera configuration table.

```
#define D_ELIB_CAM_TYPE_FRONT           /* Front camera */
#define D_ELIB_CAM_TYPE_SIDE            /* Back camera */
#define D_ELIB_CAM_MODE_CONTACT         /* Burst mode */
#define D_ELIB_CAM_MODE_PORTRAIT        /* Portrait mode */
#define D_ELIB_CAM_MODE_SCENERY         /* Scenery mode */
#define D_ELIB_CAM_MODE_NORMAL          /* Standard */
#define D_ELIB_CAM_MODE_NIGHT           /* Night mode */
#define D_ELIB_CAM_MODE_OCR             /* OCR mode */
#define D_ELIB_CAM_MODE_BARCODE         /* Two-dimensional bar code mode */
#define D_ELIB_CAM_MODE_SPORTS              /* Sports mode */
                        /* Return value */
#define D_ELIB_CAM_DRIVER_NG            /* Driver error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Get_Photo_Mode(unsigned int ap_id, int camera_type)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	int	In	Camera type Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE
Return value	**Type**	**I/O**	**Description**
Ret	int	Out	Photographing mode Close-up : D_ELIB_CAM_MODE_CONTACT Portrait : D_ELIB_CAM_MODE_PORTRAIT Scenery : D_ELIB_CAM_MODE_SCENERY (Standard: D_ELIB_CAM_MODE_NORMAL) * Equivalent to Scenery (to match text in the operation specifications) Night mode : D_ELIB_CAM_MODE_NIGHT OCR : D_ELIB_CAM_MODE_OCR

			Two-dimensional bar code : D_ELIB_CAM_MODE_BARCODE Sports mode: D_ELIB_CAM_MODE_SPORTS Driver error : D_ELIB_CAM_DRIVER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resource not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The data to be read is stored in the area corresponding to the pertinent camera type, of camera configuration table.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.36 Burst shot interval setting status read

Classification	Camera service ELIB function		
Function name	Burst shot function setting	Symbol	Elib_CAM_Get_Burst_Speed
Function overview			

Reads the burst shot interval setting from non-volatile memory if the recording style is burst shooting.

```
#define D_ELIB_CAM_BURST_SPEED_SLOW       /*  2.0 seconds */
#define D_ELIB_CAM_BURST_SPEED_QUICK      /*  1.0 second */
#define D_ELIB_CAM_BURST_SPEED_MACH       /*  0.5 second */
                                  /* Return value */
#define D_ELIB_CAM_DRIVER_NG              /* Driver error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG     /* Resources not acquired */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Get_Burst_Speed(unsigned int ap_id)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	**Type**	**I/O**	**Description**
Ret	Int	Out	Burst shot interval 2.0 seconds: D_ELIB_CAM_BURST_SPEED_SLOW 1.0 second: D_ELIB_CAM_BURST_SPEED_QUICK 0.5 second: D_ELIB_CAM_BURST_SPEED_MACH Driver error : D_ELIB_CAM_DRIVER_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The data to be read is stored in non-volatile memory.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.37 Camera zoom information read

Classification	Camera service ELIB function		
Function name	Camera zoom information read	Symbol	Elib_CAM_Get_Zoom_Info
Function overview			
Reads the zoom information for the specified camera device. #define D_ELIB_CAM_TYPE_FRONT /* Front camera */ #define D_ELIB_CAM_TYPE_SIDE /* Back camera */ /* Return value */ #define D_ELIB_CAM_OK /* Normal end */ #define D_ELIB_CAM_HANDLER_NG /* Handler error */ #define D_ELIB_CAM_PARAM_NG /* Parameter error */ #define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */ #define D_ELIB_CAM_POWER_OFF_NG /* Camera power off */ #define D_ELIB_CAM_OTHER_CTL_ACTION_NG /* Other control action in progress */ #define D_ELIB_CAM_OTHERS_NG /* Other error */			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Get_Zoom_Info(unsigned int ap_id, int camera_type)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	int	In	Camera type Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE
Return value	Type	I/O	Description
Ret	int	Out	Normal end : D_ELIB_CAM_OK Handler error: D_ELIB_CAM_HANDLER_NG Parameter error : D_ELIB_CAM_PARAM_NG Resources not acquired : D_ELIB_CAM_RESOURCE_NOGAIN_NG Camera power off : D_ELIB_CAM_POWER_OFF_NG Other control action in progress : D_ELIB_CAM_OTHER_CTL_ACTION_NG Other error : :D_ELIB_CAM_OTHERS_NG
Remarks			
- The camera zoom information is reported as a zoom information notification event in non-synchronization with the return of this function.			

- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If an attempt is made to read zoom information of a camera device on which other control action is being performed, this function returns abnormally with an Other control action in progress error.
- If an attempt is made to read zoom information of a camera device whose power is off, this function returns abnormally with a Camera power off error.

Middle layer interface:
- [Middle Layer] function that obtains the current number of zoom steps ()
- [Middle Layer] callback function that reports the current number of zoom steps.

1.38 Camera off timer setting status read (not used)

<table>
<tr><td>Classification</td><td colspan="4">Camera service ELIB function</td></tr>
<tr><td>Function name</td><td colspan="2">Camera off timer setting status read</td><td>Symbol</td><td>Elib_CAM_Get_OffTimer</td></tr>
<tr><td>Function overview</td><td colspan="4"></td></tr>
<tr><td colspan="5">Reads the camera auto off timer setting from non-volatile memory.

#define D_ELIB_CAM_OFF_TIMER_ON /* Camera off timer enabled */
#define D_ELIB_CAM_OFF_TIMER_OFF /* Camera off timer disabled */
/* Return value */
#define D_ELIB_CAM_OK /* Normal end */
#define D_ELIB_CAM_DRIVER_NG /* Driver error */
#define D_ELIB_CAM_PARAM_NG /* Parameter error */</td></tr>
<tr><td>Include File</td><td colspan="4">srv_cam.h</td></tr>
<tr><td>Calling Sequence</td><td colspan="4">int Elib_CAM_Get_OffTimer(unsigned int ap_id, T_ELIB_CAM_OFF_TIMER *timer)</td></tr>
<tr><td>Argument name</td><td>Type</td><td>I/O</td><td colspan="2">Description</td></tr>
<tr><td>ap_id</td><td>unsigned int</td><td>In</td><td colspan="2">Application ID</td></tr>
<tr><td>*timer</td><td>T_ELIB_CAM_OFF_TIMER</td><td>In/Out</td><td colspan="2">Pointer to the camera off time setting data structure
typedef struct t_elib_cam_off_timer {
int on_off; /* Off timer setting
Timer enabled: D_ELIB_CAM_OFF_TIMER_ON
Timer disabled: D_ELIB_CAM_OFF_TIMER_OFF
int timer; /* Timer value (1 to 60 minutes) */
} T_ELIB_CAM_OFF_TIMER;
* If the timer is disabled, the specified timer value will be treated as invalid.</td></tr>
<tr><td>Return value</td><td>Type</td><td>I/O</td><td colspan="2">Description</td></tr>
<tr><td>Ret</td><td>int</td><td>Out</td><td colspan="2">Normal end : D_ELIB_CAM_OK
Driver error : D_ELIB_CAM_DRIVER_NG
Parameter error: D_ELIB_CAM_PARAM_NG</td></tr>
<tr><td>Remarks</td><td colspan="4"></td></tr>
<tr><td colspan="5">- The data to be read is stored in non-volatile memory.
- The initial settings are timer enabled and a timer value of 5 minutes.
- Since it is assumed that the camera off timer setting is read on the overall configuration screen, the</td></tr>
</table>

off timer can be used regardless of whether camera resources are acquired.

1.39 Record size setting status read

<table>
<tr><td>Classification</td><td colspan="4">Camera service ELIB function</td></tr>
<tr><td>Function name</td><td>Record size setting status read</td><td>Symbol</td><td colspan="2">Elib_CAM_Get_Record_Size</td></tr>
<tr><td>Function overview</td><td colspan="4"></td></tr>
<tr><td colspan="5">Reads the record size setting from non-volatile memory.

#define D_ELIB_CAM_REC_SIZE_MAIL_ATTACH /* Moving picture: Mail */
#define D_ELIB_CAM_REC_SIZE_NO_LIMIT /* Moving picture: Moving picture memo */
#define D_ELIB_CAM_REC_SIZE_LONG_TIME /* Moving picture: Long time */
/* Return value */
#define D_ELIB_CAM_DRIVER_NG /* Driver error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */</td></tr>
<tr><td>Include File</td><td colspan="4">srv_cam.h</td></tr>
<tr><td>Calling Sequence</td><td colspan="4">int Elib_CAM_Get_Record_Size(unsigned int ap_id)</td></tr>
<tr><td>Argument name</td><td>Type</td><td>I/O</td><td colspan="2">Description</td></tr>
<tr><td>ap_id</td><td>unsigned int</td><td>In</td><td colspan="2">Application ID</td></tr>
<tr><td>Return value</td><td>Type</td><td>I/O</td><td colspan="2">Description</td></tr>
<tr><td>Ret</td><td>int</td><td>Out</td><td colspan="2">Record size
- Moving picture record size
Mail : D_ELIB_CAM_REC_SIZE_MAIL_ATTACH
Moving picture memo : D_ELIB_CAM_REC_SIZE_NO_LIMIT
Long time: D_ELIB_CAM_REC_SIZE_LONG_TIME
Driver error : D_ELIB_CAM_DRIVER_NG
Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG</td></tr>
<tr><td>Remarks</td><td colspan="4"></td></tr>
<tr><td colspan="5">- The data to be read is stored in non-volatile memory.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.</td></tr>
</table>

1.40 Recording medium setting status read

Classification	Camera service ELIB function		
Function name	Camera recording medium setting status read	Symbol	Elib_CAM_Get_Record_Media
Function overview			
Reads the recording medium setting corresponding to the specified recording mode, from the camera configuration table. #define D_ELIB_CAM_SNAP /* Still picture */ #define D_ELIB_CAM_MOVIE /* Moving picture */ #define D_ELIB_CAM_RECORD_INSIDE /* Internal memory */ /* Return value */ #define D_ELIB_CAM_DRIVER_NG /* Driver error */ #define D_ELIB_CAM_PARAM_NG /* Parameter error */ #define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resources not acquired */			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Get_Record_Media(unsigned int ap_id, int record_mode)		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE
Return value	Type	I/O	Description
Ret	int	Out	Recording medium flag Internal memory: D_ELIB_CAM_RECORD_INSIDE Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG
Remarks			
- The data to be read is stored in the area corresponding to the pertinent recording mode, of camera configuration table. - If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.			

1.41 Recording quality setting status read

Classification	Camera service ELIB function		
Function name	Recording quality setting status read	Symbol	Elib_CAM_Get_Record_Quality
Function overview			

Reads the image recording quality setting from the area corresponding to the pertinent recording mode (still or moving) and the recording media (inside/SD), of non-volatile memory.

```
#define D_ELIB_CAM_SNAP                 /* Still picture */
#define D_ELIB_CAM_MOVIE                /* Moving picture */
#define D_ELIB_CAM_RECORD_INSIDE        /* Internal memory */
#define D_ELIB_CAM_QUALITY_ECONOMY      /* Still picture: Mail restriction; Moving picture: Priority on time */
#define D_ELIB_CAM_QUALITY_NORMAL       /* Still picture: Large-size mail restriction; Moving picture: Standard */
#define D_ELIB_CAM_QUALITY_FINE         /* Still picture: No restriction; Moving picture: Priority on picture quality */
                                        /* Return value */
#define D_ELIB_CAM_DRIVER_NG            /* Driver error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Get_Record_Quality(unsigned int ap_id, int record_mode, int media)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE
Media	int	In	Recording medium flag Internal memory: D_ELIB_CAM_RECORD_INSIDE
Return value	Type	I/O	Description
Ret	int	Out	Recording quality flag Mail restriction (still pictures); Priority on time (moving pictures) : D_ELIB_CAM_QUALITY_ECONOMY Large-size mail restriction (still pictures); Standard (moving pictures) : D_ELIB_CAM_QUALITY_NORMAL No restriction (still pictures); Priority on picture quality (moving pictures)

			: D_ELIB_CAM_QUALITY_FINE Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The data to be read is stored in the area corresponding to the pertinent recording mode and the recording medium flag, of volatile memory.
However, if the recording mode is Still, the recording medium flag is invalid.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- In Still picture mode, the recording quality flag that is reported is Mail restriction, Large-size mail restriction, or No restriction.
- In Moving picture mode, the recording quality flag that is reported is Priority on time, Standard, or Priority on picture.

1.42 Reading of various camera settings

Classification	Camera service ELIB function		
Function name	Reads various camera settings.	Symbol	Elib_CAM_Get_General_Conf
Function overview			
Reads various general settings to be used for camera applications, from non-volatile memory. The following settings can be read: typedef struct tagELIB_CAM_GENERALCONF { BYTE bSelfTimer Self-timer duration in seconds BYTE bShutSound_ST Still picture/burst shot shutter sound BYTE bShutSound_MV Moving picture shutter sound BYTE bBurstMode Burst shooting type BYTE bAutoSave Automatic registration setting BYTE bDispSize: View size } _ELIB_CAM_GENERALCONF;			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Get_General_Conf (unsigned int ap_id, _ELIB_CAM_GENERALCONF* pConf);		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
pConf	_ELIB_CAM_GENERALCONF*	In	Information structure for various camera settings
Return value	Type	I/O	Description
Ret	Int	Out	Normal end : D_ELIB_CAM_OK Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG
Remarks			
- The data to be read is stored in non-volatile memory. - If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.			

1.43 Reading of default start-up camera settings

Classification	Camera service ELIB function		
Function name	Default start-up camera settings read	Symbol	Elib_CAM_Get_Def_Camera
Function overview			
Reads the default start-up camera setting from non-volatile memory.			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Get_Def_Camera (unsigned int ap_id);		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	Type	I/O	Description
Ret	int	Out	Default camera Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE Driver error : D_ELIB_CAM_DRIVER_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG
Remarks			

- The data to be read is stored in non-volatile memory.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.44 Camera mode dependent default zoom value read

Classification	Camera service ELIB function		
Function name	Camera mode dependent default zoom value read	Symbol	Elib_CAM_Get_Def_Zoom
Function overview			

Reads the initial zoom value that corresponds to the camera type (front or back) and the recording mode (Still, Moving, or Burst).
The camera mode dependent default zoom value table is initialized to no-zoom (0) when the camera resources are acquired.
If the recording resolution is changed by a call to Elib_CAM_Set_Record_Resolution or Elib_CAM_SetRecord_Resolution_All, the zoom value corresponding to that camera mode is initialized to no-zoom (0).
The camera mode dependent default zoom value table is read when the power is turned on by Elib_CAM_Ctrl_Camera_Active or when the recording mode is changed by Elib_CAM_Ctrl_Record_Mode_Change.
The camera device will be initialized with the camera mode dependent zoom value that is stored in this table.

Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Get_Def_Zoom (unsigned int ap_id, int camera_type, int record_mode);		
Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
camera_type	int	In	Camera type Front camera: D_ELIB_CAM_TYPE_FRONT Back camera: D_ELIB_CAM_TYPE_SIDE
record_mode	int	In	Recording mode Still picture: D_ELIB_CAM_SNAP Moving picture: D_ELIB_CAM_MOVIE Burst: D_ELIB_CAM_BURST_SNAP
Return value	Type	I/O	Description
Ret	int	Out	Camera mode dependent default zoom value Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- If the function is called without resource acquirement, it returns abnormally with a Resources not

acquired error.

1.45 Burst shot count setting status read

Classification	Camera Elib function		
Function name	Burst shot function setting	Symbol	Elib_CAM_Get_SeqShoot_Numbers
Function overview			
Reads the burst shot count setting from non-volatile memory. /* Return value */ #define D_ELIB_CAM_DRIVER_NG /* Driver error */ #define D_ELIB_CAM_PARAM_NG /* Parameter error */ #define D_ELIB_CAM_RESOURCE_NOGAIN_NG /* Resource not acquired */			
Include File	srv_cam.h		
Calling Sequence	int Elib_CAM_Get_SeqShoot_Numbers (unsigned int Ap_id)		

Argument name	Type	I/O	Description
Ap_id	unsigned int	In	Application ID

Return value	Type	I/O	Description
Ret	int	Out	Number of burst shots : 4 to 20 (D_ELIB_CAM_SEQSHTNUM_MIN - D_ELIB_CAM_SEQSHTNUM_MAX) Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The data to be read is stored in non-volatile memory.
- The initial value is set at 5.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.

1.46 Flicker suppression setting status read

Classification	Camera Elib function		
Function name	Flicker suppression setting status read	Symbol	Elib_CAM_Get_Flicker
Function overview			

Reads the flicker suppression setting from non-volatile memory.

```
/* Return value */
#define D_ELIB_CAM_DRIVER_NG            /* Driver error */
#define D_ELIB_CAM_PARAM_NG             /* Parameter error */
#define D_ELIB_CAM_RESOURCE_NOGAIN_NG   /* Resources not acquired */
```

Include File	srv_cam.h
Calling Sequence	int Elib_CAM_Get_Flicker (unsigned int ap_id)

Argument name	Type	I/O	Description
ap_id	unsigned int	In	Application ID
Return value	Type	I/O	Description
Ret	int	Out	Flicker suppression setting Auto: D_ELIB_CAM_FLICKER_AUTO 50 Hz: D_ELIB_CAM_FLICKER_50 60 Hz: D_ELIB_CAM_FLICKER_60 Driver error : D_ELIB_CAM_DRIVER_NG Parameter error: D_ELIB_CAM_PARAM_NG Resources not acquired: D_ELIB_CAM_RESOURCE_NOGAIN_NG

Remarks

- The data to be read is stored in non-volatile memory.
- The initial value is set at Auto.
- If the function is called without resource acquirement, it returns abnormally with a Resources not acquired error.
- If the flicker suppression setting that is read from non-volatile memory is not Auto, 50 Hz, or 60 Hz, this function returns abnormally with a Driver error.

2. Video Control Service
2.1 Initial Processing

Classification	Support Functions of the Video Control Service		
Function Name	Initial Process	Symbol	Elib_MS_Initialize

Functional Overview

The AP that uses the Video Control ELIB calls this function during activation.
By executing this function, the Video Service can be received.(The Application ID is the access key.)
Secure the area (Event management, Attribute Information, Video Information returned to AP) of the information that manages the AP that uses Video Control ELIB.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_Initialize (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
Ret	int	O	[Normal Completion] Normal Completion: ELIB_MS_OK [Abnormal Completion] Abnormal Completion: ELIB_MS_NG

Remark

- Register to the Application Management Table within the Video Control Service.
(Secure area to store the Event, Attribute and Video Information for each AP.)

- For requests from the same application, return ELIB_MS_NG as an error.

The Application ID that is the input parameter can be set optionally, but it is necessary to make sure the same value is not used between the applications in the system.(It is recommended that values such as Program ID and Task ID are used.)

- Acquire the authority to execute this function and receive the service during Application Activation.
Release the authority to execute the Complete Process and receive the service during Application Completion.(If it is not released, the resource is not released in Video ELib and there is a possibility of insufficient memory.Also, Initialization Processes using the same Ap_ID will fail.)

When the power is ON, the application that uses the Video Control Service conducts initialization when necessary and end it when it is no longer necessary.In use methods such as when initialization is conducted when the power is ON and completion when the power is OFF, the various memory areas acquired through initialization are wasted.

2.2 End Processing

Classification	Support function for video control service		
Function	Termination processing	Symbol	Elib_MS_Terminate
Functional overview			
Waive the right to receive movie services acquired by the initialization. The movie service that specifies the same AP_ID after the termination processing will generate an error. Termination processing will release the information area that manages AP acquired in the initialization.			
Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Terminate (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG
Note			
- For ELIB_MS_Terminate while it has not conducted Elib_MS_Initialize, it will make normal end. (It is not processed internally.) - Basically, while the movie control service is in processing (until event return after the request etc.), the termination processing will make abnormal end and will not become effective. In that case, retry after terminating the movie control service processing.			

2.3 Event Occurrence Notification Request

Classification	Support Functions of the Video Replay Service		
Function Name	Event Occurrence Notification Request	Symbol	Elib_MS_Request

Functional Overview

Sets the notification request of events that occur voluntarily from the Video Service and the result event during various Service Requests.
Multiple designations are possible as the same time for Events due to logical sum.

By executing this function, the events that occur due to the Video Service Opearations that follow can be received.

When requesting events such as Video Replay, Video Record, Video Edit, Still Image Record, Video Registration and 6 other types of events, it is necessary to call this function 6 times.

Include file	srv_ms_p.h
Calling Sequence	int Elib_MS_Request(unsigned int Ap_ID, int event, MsbFunc CallBack_ADDR);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Event	int	I	Events to be notified (more than one can be designated due to logical sum within each Event Type) * Type: Video Replay, Video Record, Video Edit, Still Image Record, Video Registration, others [Video Replay] MSNotify_PLAY_END : Replay Complete MSNotify_PLAY_FPICT : First Image Replay Complete Notification MSNotify_PLAY_PAUSE : Pause Process Complete MSNotify_PLAY_SEEK : Seek Complete MSNotify_PLAY_STEP : Step/Frame Advance Complete MSNotify_PLAY_STOP : Stop Complete MSNotify_PLAY_TIME : Replay Time Notification MSNotify_PLAY_SPAUSE : Data Inadequate Pause MSNotify_PLAY_SRESUME: Data Inadequate Pause Resume MSNotify_PLAY_ERROR : Replay Error MSNotify_PLAY_QTTDATA : Quick Time Text Data MSNotify_PLAY_RELDATA : Acquired Data Release Notification [Video Record] MSNotify_REC_START : Record Start MSNotify_REC_END : Record Complete (Complete Condition Established) MSNotify_REC_DONE : Record Complete (Post-processing Complete) MSNotify_REC_TIME : Record Time Notification MSNotify_REC_ERROR : Record Error MSNotify_REC_CANCEL: Record Cancellation Complete MSNotify_REC_REMAIN : Video Record Remaining Time Notification

			[Video Edit] MSNotify_EDIT_START : Edit Start MSNotify_EDIT_FPICT : Edit First Image Replay Complete Notification MSNotify_EDIT_END : Edit Complete (Complete Condition Established) MSNotify_EDIT_DONE : Edit Complete (Post-processing Complete) MSNotify_EDIT_TIME : Edit Status Notification MSNotify_EDIT_ERROR : Edit Error MSNotify_EDIT_CANCEL: Edit Cancellation Complete [Still Image Record] MSNotify_SNAP_DONE : Still Image Record Complete MSNotify_SNAP_ERROR : Still Image Record Error MSNotify_SNAP_CANCEL: Still Image Record Cancellation Complete [Video Registration] MSNotify_SET_END : Video Data Registration Complete MSNotify_SET_ERROR : Video Data Registration Failure MSNotify_SET_CANCEL: Video Data Registration Cancellation Complete MSNotify_GET_DONE : Video Data Read Complete MSNotify_GET_ERROR : Video Data Read Failure MSNotify_GET_CANCEL: Video Data Read Cancellation Complete [Other] MSNotify_OTHER_ERROR: Other errors [Multiple Designation] MSNotify_PLAY_ALL : Set all Video Replay events MSNotify_REC_ALL : Set all Video Record events MSNotify_EDIT_ALL : Set all Video Edit events MSNotify_SNAP_ALL : Set all Still Image Record events MSNotify_SET_ALL : Set all Video Registration events MSNotify_OTHER_ALL : Set all other events
CallBack_ADDR	MsbFunc	I	Call Back Function Address
Return value	Type	-	Description
Ret	int	O	[Normal Completion] Normal Completion: ELIB_MS_OK [Abnormal Completion] Abnormal Completion: ELIB_MS_NG

Remark

- Concerning the description of each event, refer to "3. Event".
Only one Call Back function address can be registered for each Ap_ID.
- If the Event Occurance Notification Request is conducted several times, the last Call Back Function Address to be registered is valid.

2.4 Event Occurance Notification Release

Classification	Support Functions of the Video Control Service		
Function Name	Event Occurrence Notification Release	Symbol	Elib_MS_Cancel

Functional Overview

Sets notification not necessary for events that occur voluntarily from the Video Service and the result event during various Service Requests.
Multiple designations are possible as the same time for Events due to logical sum.

By executing this function, the events that occur due to the Video Service Opearations are no longer notified.

When releasing events such as Video Replay, Video Record, Video Edit, Still Image Record, Video Registration and 6 other types of events, it is necessary to call this function 6 times.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_Cancel(unsigned int Ap_ID , int event);		
Argument	**Type**	**I/O**	**Description**
Ap_ID	Unsigned int	I	Application ID
Event	Int	I	Events that are not to be notified (more than one can be designated due to logical sum within each Event Type) Type: Video Replay, Video Record, Video Edit, Still Image Record, Video Registration, others [Video Replay] MSNotify_PLAY_END : Replay Complete MSNotify_PLAY_FPICT : First Image Replay Complete Notification MSNotify_PLAY_PAUSE : Pause Process Complete MSNotify_PLAY_SEEK : Seek Complete MSNotify_PLAY_STEP : Step/Frame Advance Complete MSNotify_PLAY_STOP : Stop Complete MSNotify_PLAY_TIME : Replay Time Notification MSNotify_PLAY_SPAUSE : Data Inadequate Pause MSNotify_PLAY_SRESUME: Data Inadequate Pause Resume MSNotify_PLAY_ERROR : Replay Error MSNotify_PLAY_QTTDATA : Quick Time Text Data MSNotify_PLAY_RELDATA : Acquired Data Release Notification [Video Record] MSNotify_REC_START : Record Start MSNotify_REC_END : Record Complete (Complete Condition Established) MSNotify_REC_DONE : Record Complete (Post-processing Complete) MSNotify_REC_TIME : Record Time Notification MSNotify_REC_ERROR : Record Error MSNotify_REC_CANCEL: Record Cancellation Complete MSNotify_REC_REMAIN : Video Record Remaining Time Notification

			[Video Edit] MSNotify_EDIT_START : Edit Start MSNotify_EDIT_FPICT : Edit First Image Replay Complete Notification MSNotify_EDIT_END : Edit Complete (Complete Condition Established) MSNotify_EDIT_DONE : Edit Complete (Post-processing Complete) MSNotify_EDIT_TIME : Edit Status Notification MSNotify_EDIT_ERROR : Edit Error MSNotify_EDIT_CANCEL: Edit Cancellation Complete [Still Image Record] MSNotify_SNAP_DONE : Still Image Record Complete MSNotify_SNAP_ERROR : Still Image Record Error MSNotify_SNAP_CANCEL: Still Image Record Cancellation Complete [Video Registration] MSNotify_SET_END : Video Data Registration Complete MSNotify_SET_ERROR : Video Data Registration Failure MSNotify_SET_CANCEL : Video Data Registration Cancellation Complete MSNotify_GET_DONE : Video Data Read Complete MSNotify_GET_ERROR : Video Data Read Failure MSNotify_GET_CANCEL: Video Data Read Cancellation Complete [Other] MSNotify_OTHER_ERROR: Other errors [Multiple Designation] MSNotify_PLAY_ALL : Release all Video Replay events MSNotify_REC_ALL : Release all Video Record events MSNotify_EDIT_ALL : Release all Video Edit events MSNotify_SNAP_ALL : Release all Still Image Record events MSNotify_SET_ALL : Release all Video Registration events MSNotify_OTHER_ALL : Release all other events
Return value	Type	-	Description
ret	Int	O	[Normal Completion] Normal Completion: ELIB_MS_OK [Abnormal Completion] Abnormal Completion: ELIB_MS_NG
Remark			

- Concerning the description of each event, refer to "3. Event".

Due to the problem of Task Priority, there are cases where notification is possible immediately after release.The process to eliminate the relevant event is also necessary as an application after release.

2.5 Acquisition Of Service Attributes

Classification	Support function for video control service		
Function	Get service attributes	Symbol	Elib_MS_GetProperty
Functional overview			

Get one of the service attributes of video control service. (Read out)
Service attribute is an operation parameter that will become the base when movie play/movie recording/edit operation/ still image recording are executed.
Service attribute is defined for each ApplicationID.

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_GetProperty(unsigned int Ap_ID , int Prop_ID, void *Prop_Val);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Prop_ID	int	I	Identifier of the attribute to get (specify only one) [Play-related] MSProp_PLAY_INT : Interval of time for notifying play time MSProp_PLAY_BGC : Background color for the movie playing [Movie recording-related] MSProp_REC_INT : Interval of time for notifying recording time MSProp_REC_LIMITSIZE : Recording capacity limit MSProp_REC_LIMITTIME : Recording time limit MSProp_REC_FORMAT : File format during recording MSProp_REC_VQUALITY : Codec/image quality during recording MSProp_REC_AQUALITY : Codec/audio quality during recording MSProp_REC_DIST_RIGHT: Information for the recognition of redistribution during recording MSProp_REC_TITLE : Title during recording [Movie editing-related] MSProp_EDIT_INT : Interval of time for notifying editing status MSProp_EDIT_BGC : Background color for the movie editing MSProp_EDIT_LIMITSIZE : Editing capacity limit MSProp_EDIT_LIMITTIME : Editing time limit MSProp_EDIT_VQUALITY : Codec/image quality during editing MSProp_EDIT_AQUALITY : Codec/audio quality during editing MSProp_EDIT_DIST_RIGHT: Information for the recognition of redistribution during editing

			MSProp_EDIT_TITLE : Title during editing [Still image recording-related] MSProp_SNAP_SQUALITY: Format/image quality for still image
Prop_Val	void *	O	Pointer to corresponding data by each identifier
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Normal end] Abnormal end :ELIB_MS_NG

Remark

- Memory for reading service attributes should be freed up by application. (Note that the size varies by type)
- For the result of retrieval, retrieve information by the type conformed to service attribute.

2.6 Total Acquisition Of Service Attributes

Classification	Support Functions of the Video Control Service		
Function Name	Service Attribute All Acquisition	Symbol	Elib_MS_GetAllProperty
Functional Overview			

- Get all service attributes or the service attributes of each category of the Video Control Service (except the Video Edit Service).

- The Service Attribute is the Operation Parameter that becomes the base for when Video Replay/Video Record/Still Image Record is conducted.
- The Service Attribute is defined for each Application ID.
- If MSProp_ALL (All Service Attribute Acquisition) is designated in Prop_ID, this interface cannot conduct all acquisition of the Video Edit Service Attribute.Video Edit All Service Attribute Setting Data Acquisitionis used.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetAllProperty(unsigned int Ap_ID , int Prop_ID, _ELIB_MS_PLAYPROP* PlayProp_Val, _ELIB_MS_RECPROP* RecProp_Val, _ELIB_MS_SNAPPROP* SnapProp_Val);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Prop_ID	Int	I	Identifier of the attribute to be acquired (designate only one) MSProp_PLAY : Video Replay Service Attribute All Acquisition MSProp_REC : Video Record Service Attribute All Acquisition MSProp_SNAP : Still Image Record Service Attribute All Acquisition MSProp_ALL : All Service Attribute Acquisition
PlayProp_Val	_ELIB_MS_PLAYPROP *	O	Video Replay All Service Attribute Setting Data
RecProp_Val	_ELIB_MS_RECPROP *	O	Video Record All Service Attribute Setting Data
SnapProp_Val	_ELIB_MS_SNAPPROP*	O	Still Image Record All Service Attribute Setting Data
Return value	Type	-	Description
ret	Int	O	[Normal Completion] Normal Completion: ELIB_MS_OK [Abnormal Completion] Abnormal Completion: ELIB_MS_NG
Remark			

- The memory that reads the Service Attribute is secured in the Application.(Note that the size differs

depending on the type.

- The extracted result should be extracted in accordance to the Service Attribute.

- If MSProp_PLAY/REC/SNAP is designated, only prepare the memory that acquires the corresponding setting values.Set the Read out destination of the other setting values at "NULL".
  - If MSProp_ALL is designated, prepare a memory to acquire all setting values.

- Concerning the Service Attributes of the Video Record, also refer to "3. Event".

2.7 Service Attributes Setting

Classification	Support function for video control service		
Function	Service attributes setting	Symbol	Elib_MS_SetProperty
Functional overview			

Set one of the service attributes of video control service.
Service attribute is an operation parameter that will become the base when movie play/movie recording/edit operation/still image recording are executed.
Service attribute is defined for each ApplicationID.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_SetProperty(unsigned int Ap_ID , int Prop_ID, void *Prop_Val);		
Argument	Type	I/O	Description
Ap_ID	Unsigned int	I	Application ID
Prop_ID	Int	I	Identifier of the attribute to be set (specify only one) [Movie play-related] MSProp_PLAY_INT : Interval of time for notifying play time MSProp_PLAY_BGC : Background color for the movie playing [Movie recording-related] MSProp_REC_INT : Interval of time for notifying recording time MSProp_REC_LIMITSIZE : Recording capacity limit MSProp_REC_LIMITTIME : Recording time limit MSProp_REC_FORMAT : File format during recording MSProp_REC_VQUALITY : Codec/image quality during recording MSProp_REC_AQUALITY : Codec/audio quality during recording MSProp_REC_DIST_RIGHT: Information for the recognition of redistribution during recording MSProp_REC_TITLE : Title during recording [Movie editing-related] MSProp_EDIT_INT : Interval of time for notifying editing status MSProp_EDIT_BGC : Background color for the movie editing MSProp_EDIT_LIMITSIZE : Editing capacity limit MSProp_EDIT_LIMITTIME : Editing time limit MSProp_EDIT_VQUALITY : Codec/image quality during editing MSProp_EDIT_AQUALITY : Codec/audio quality during editing MSProp_EDIT_DIST_RIGHT: Information for the recognition of redistribution during editing MSProp_EDIT_TITLE : Title during editing

			[Still image recording-related] MSProp_SNAP_SQUALITY: Format/image quality for still image
Prop_Val	void *	I	Pointer to corresponding data by each identifier
Return value	Type	-	Description
ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG
Remark			

- If play-related service attributes are set while the movie is playing, the service attributes will become valid in the next time the movie plays.
- If movie recording-related service attributes are set while recording the movie, the service attributes will become valid in the next time you record the movie.
- If movie editing-related service attributes are set while editing the movie, the service attributes will become valid in the next time you edit the movie.
- If still image recording-related service attributes are set while recording the still image, the service attributes will become valid in the next time you record the still image.
- Set information by the type conformed to service attribute.

2.8 Total Setting Of Service Attributes

Classification	Support Functions of the Video Control Service		
Function Name	Service Attribute All Configuration	Symbol	Elib_MS_SetAllProperty
Functional Overview			

- Set all service attributes or the service attributes of each category of the Video Control Service (except the Video Edit Service).
- The Service Attribute is the Operation Parameter that becomes the base for when Video Replay/Video Record/Still Image Record is conducted.
- The Service Attribute is defined for each Application ID.
- If MSProp_ALL (All Service Attribute Setting) is designated in Prop_ID, this interface cannot conduct all settings of the Video Edit Service Attribute.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_SetAllProperty(unsigned int Ap_ID , int Prop_ID, _ELIB_MS_PLAYPROP* PlayProp_Val, _ELIB_MS_RECPROP* RecProp_Val, _ELIB_MS_SNAPPROP* SnapProp_Val);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Prop_ID	Int	I	Identifier of the attribute to be set (designate only one) MSProp_PLAY : Video Replay Service Attribute All Setting MSProp_REC : Video Record Service Attribute All Setting MSProp_SNAP : Still Image Record Service Attribute All Setting MSProp_ALL : All Service Attribute Setting
PlayProp_Val	_ELIB_MS_PLAYPROP	I	Video Replay All Service Attribute Setting Data
RecProp_Val	_ELIB_MS_RECPROP	I	Video Record All Service Attribute Setting Data
SnapProp_Val	_ELIB_MS_SNAPPROP	I	Still Image Record All Service Attribute Setting Data
Return value	Type	-	Description
ret	Int	O	[Normal Completion] Normal Completion: ELIB_MS_OK [Abnormal Completion] Abnormal Completion: ELIB_MS_NG

Remark

- If the Service Attribute of the Replay is set during Replay, the next time the Service Attribute is valid is during the next Replay.

- If the Service Attribute of the Video Record is set during Video Recor, the next time the Service Attribute is valid is during the next Video Record.

- If the Service Attribute of the Still Image Record is set during Still Image Record, the next time the Service Attribute is valid is during the next Still Image Record.

- If MSProp_PLAY/REC/SNAP is designated, prepare the corresponding setting values.
  Set the other setting values at "NULL".
  - If MSProp_ALL is designated, prepare all setting values.

- Concerning the Service Attributes of the Video Record, also refer to "3. Event".

2.9 Resetting Of Service Attributes

Classification	Support function for video control service		
Function	Service attributes reset	Symbol	Elib_MS_ResetProperty
Functional overview			

Restore all the service attributes to be set for each AP to the default value.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_ResetProperty(unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Configure the default value for service attributes by the constant value in the header file in implementing the movie Elib. (It is determined in creating executable file.)
- In the case of becoming the default value while the movie is playing, the service attributes will become valid in the next time the movie plays.
- In the case of becoming the default value while recording the movie, the service attributes will become valid in the next time you record the movie.
- In the case of becoming the default value while editing the movie, the service attributes will become valid in the next time you edit the movie.
- In the case of becoming the default value while recording the still image, the service attributes will become valid in the next time you record the still image.

2.10 Movie Play Display at the Same Magnification/Enlarged Setting Reference Processing

Classification	Support function for video control service		
Function	Movie play display at the same magnification/enlarged setting reference processing	Symbol	Elib_MS_SetRef_DispSize
Functional overview			

- Processing for recording the setting of moving image display (at the same magnification/enlarged).
- From Specify the operation (Mode), select Setting Mode/Reference Mode.
- If the pointer argument is NULL, it returns Parameter abnormal.
- If the argument is other than the specified value, it returns Parameter abnormal.

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_SetRef_DispSize (unsigned int Ap_ID, int Mode, int *Value);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	Int	I	[Specify the operation] ELIB_MS_SET : Setting mode ELIB_MS_REF : Reference mode
Value	int*	I/O	[Image display setting value] ELIB_MS_STANDARD : Display at the same magnification ELIB_MS_LARGE : Display enlarged
Return value	**Type**	**-**	**Description**
ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark
Nothing in particular

2.11 Movie Data Registration

Classification	Support Functions of the Video Control Service		
Function Name	Video Data Registration	Symbol	Elib_MS_SetClip
Functional Overview			

- Register the Video Data that exists in the memory of the CPU address space as a file in the FLASH.
- The registered data creates an individual Information Area (Record) in the separately created management file (Movie File Management File).
- After being registered, the data is accessed using the file number set in the folder identifier and the return value of this function.
  INBOX, Camera or the User (1-20) folder can be designated.(However, it is possible to add folders that can be designated through folder ID addition.
  The file name during download can be designated at the same time as Registration.
- Before executing this interface, designate the Management Title Language with "Title Language Setting.

Include file	srv_ms_p.h
Calling Sequence	int Elib_MS_SetClip(Ap_ID, int Folder, int FileNum, BYTE *Data, int Size, BYTE *FileName, _ELIB_MS_LIMITINFO * LimitInfo, BYTE * URLName, _ELIB_MS_SUBINFO * SubInfo);

Argument	Type	I/O	Description
Ap_ID	Unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number to be registered (1-) If 0 is set, the smallest free number is automatically assigned in the Video ELIB.

			- Maximum Video File Value: ELIB_MS_MP4FILE_MAX Other: Assign the designated number as the file ID.
Data	BYTE*	I	Registration Source Data Top Address
Size	Int	I	Registration Source Data Size
Filename	BYTE*	I	File Name when downloading
LimitInfo	_ELIB_MS_LIMITINFO*	I	-
URLName	BYTE*	I	The URL Character String (without file name) where the Registered Contents where stored
SubInfo	_ELIB_MS_SUBINFO*	I	Registration Sub-Information (Refer to the Remarks for the setting value)
Return value	Type	-	Description
ret	Int	O	[Normal Completion] The file number within the registered folder 1- (positive value) - Video File Maximum value : ELIB_MS_MP4FILE_MAX [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG
Remark			

Files are accessed using the file number, therefore the file name is not always necessary.
However, putting into consideration situations where the file name is necessary such as when attaching to mail, the designation of the file name is possible.(The file name creates a record in the Movie Management File and manages it separately from the file names on FAT.)
If the file name is set as NULL, leave it as NULL.

If the file number to be registered is already registered or the number does not exist in the folder, return ELIB_MS_NG.

- The URL of the registered file should be "the URL where the Acquired contents are stored".Note that it is not the Content Acquisition Page or Link Destination URL (If redirect is used, Link Destination URL does not equal Contents Storage URL)
- If there are no URL Character strings that should be registered, set NULL in the top address of the Character string.
- The URL Character String secures an area that is a maximum of 256 + NULL characters.NULL should always be assigned at the end of the character string.If it exceeds 256 bytes with out detecting the NULL character, everything up to 256 bytes is valid and NULL is automatically inserted and set.(It is not an error.)
- The upper limit of the UIM ID of the Registration Contents Sub-Information is 10 bytes (ELIB_MS_UIM_CODE_MAX) and if UimID_Len is set at 11 or more, is becomes ELIB_MS_PARAM_NG.

- Register the Title Information of the language designated in the Registration Title Language as the Management Title.
- If the title of the Designated Language does not exist, set NULL at the top of the Title Character string.
- The area of the FLASH that is actually used is the total size of the Video Data Size and Management File Size.

2.12 Movie Data Registration (Asynchronous)

Classification	Support Functions of the Video Control Service		
Function Name	Video Data Registration (Asynchronous)	Symbol	Elib_MS_SetClipAsync
Functional Overview			

- Register the Video Data that exists in the memory of the CPU address space as a file in the FLASH.
- The registered data creates an individual Information Area (Record) in the separately created management file (Movie File Management File).
- After being registered, the data is accessed using the file number set in the folder and the return value of this function.
- INBOX, Camera or the User (1-20) folder can be designated.(However, it is possible to add folders that can be designated through folder ID addition.
- The file name during download and the Replay Restriction Information can be designated at the same time as Registration.

Return the result in the Data Registration Complete Event.
MSNotify_SET_END: Video Data Registration Complete
MSNotify_SET_ERROR: Video Data Registration Failure

- The Video Data Registration (Asynchronous) (Simultaneous Multiple Registration) during Video Data Registration (Asynchronous) returns error.The Video Data Registration (Asynchronous) in the Registered Video Data Read (Asynchronous) also returns error.
- Before executing this interface, designate the Management Title Language with "Title Language Setting.

Include file	srv_ms_p.h
Calling Sequence	int Elib_MS_SetClipAsync (Ap_ID, int Folder, int FileNum, BYTE *Data, int Size, BYTE *FileName, _ELIB_MS_LIMITINFO * LimitInfo, BYTE * URLName, _ELIB_MS_SUBINFO * SubInfo);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16

			ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number to be registered (1-) If 0 is set, the smallest free number is automatically assigned in the Video ELIB. - Maximum Video File Value: ELIB_MS_MP4FILE_MAX Other: Assign the designated number as the file ID.
Data	BYTE*	I	Registration Source Data Top Address
Size	Int	I	Registration Source Data Size
Filename	BYTE*	I	File Name when downloading
LimitInfo	_ELIB_MS_LIMITINFO*	I	Replay Restriction Information (Refer to "Video Data Registration" for the setting values)
URLName	BYTE*	I	The URL Character String (without file name) where the Registered Contents where stored
SubInfo	_ELIB_MS_SUBINFO*	I	Registration Contents Information (Refer to the Remarks in "Video Data Registration" for the setting values)
Return value	Type	-	Description
ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG
Remark			
For Remarks, refer to "Video Data Registration".			

2.13 Registered Movie Data Erase

Classification	Support Functions of the Video Control Service		
Function Name	Registered Video Data Deletion	Symbol	Elib_MS_DelClip
Functional Overview			

Delete the designated Registered Video Data.
The data to be deleted is the corresponding record in the management file that was defined separately from the Video File Body.
Return error if the video of the designated file number (FileNum) is not registered.
Return ELIB_MS_OK when conducting All Item Delete even if the data to be deleted does not exist.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_DelClip (unsigned int Ap_ID , int Folder, int FileNum);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number 1 Item Delete: Set File Number (1~) - Video File Maximum value: ELIB_MS_MP4FILE_MAX All Item Delete: ELIB_MS_NUMBER_ALL
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG

			Abnormal Parameter	: ELIB_MS_PARAM_NG
			Data Not Registered	: ELIB_MS_DATANOT
Remark				
None				

2.14 Acquisition Of Registered Movie Data List For Total Number

Classification	Support Functions of the Video Control Service		
Function Name	Registered Video Data List Total Acquisition	Symbol	Elib_MS_GetListNum
Functional Overview			

- Get the total number of video files (data) within the designated folder.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetListNum(unsigned int Ap_ID, int Folder);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Designate the Folder Identifier. ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U011_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U017_FOLDER: User Folder 17 ELIB_MS_U018_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] The number of Registered File Numbers: 0- - Inbox Maximum value : ELIB_MS_INBOX_MAX - Camera folder Maximum value : ELIB_MS_CAMERA_MAX - User folder Maximum value : ELIB_MS_USERF_MAX [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark
None

2.15 Acquisition Of Registered Movie Data List

Classification	Support Functions of the Video Control Service		
Function Name	Registered Video Data List Acquisition	Symbol	Elib_MS_GetMovieList
Functional Overview			

- Get the Data List of the files that were Video Data Registered within the designated folder.
The provided data is provided by conducting sort using the information in the individual information areas (record) created during Video Data Registration as a key.

Include file	srv_ms_p.h
Calling Sequence	int Elib_MS_GetMovieList(unsigned int Ap_ID, int Folder, int GetStartNumber, int GetNumber, _ELIB_MS_TITLEINFO* Title, int SortKey, int SortOrder);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
GetStartNumber	Int	I	The Number to start acquisition (The Acquisition Start Number of the data that was sorted with the sort key *It is not FileNum.)
GetNumber	Int	I	Total Number to be Acquired - The memory for the total number should be secured on the AP side.
Title	_ELIB_MS_TITLEINFO*	O	Title Information Structure
SortKey	Int	I	Sort Key Designation ELIB_MS_SORT_DATE : Acquire in the order of

			ELIB_MS_SORT_SETCLIP: Registration Date : Acquire in the order of Registration ELIB_MS_SORT_NAME : Acquire in the order of Title name ELIB_MS_SORT_MAKE : Acquire in the order of Creation Date ELIB_MS_SORT_SIZE : Acquire in the order of Data Size
SortOrder	Int	I	Designate ascending/descending order 0: Ascending order 1: Descending order
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] Number of titles acquired: 0 - (If it is 0, then the data concerned is not registered.) - Inbox Maximum value : ELIB_MS_INBOX_MAX - Camera folder Maximum value : ELIB_MS_CAMERA_MAX - User folder Maximum value : ELIB_MS_USERF_MAX [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG
Remark			

- The title sort is conducted in the order of Character code of the Character String.No distinction is made between one and two byte characters.
- The Title Information Structure of the individual GetNumber can be acquired from the GetStartNumber position of the data sorted with the sort key.
- The value designated in the GetStartNumber is 1: the top Registration Data when sorted by sort key and N: The Nth Registration Data when sorted by sort key.
- Multiple folders cannot be sorted collectively.
- The area of the Title Information Structure of GetNumber is secured on the AP side.

- The Title Information Structure is shown below.

```
typedef struct tag_ELIB_MS_TITLEINFO{
  Int                   FileNum;                                  /* File Number                                  */
  _ELIB_MS_DATE_DATA    TimeStamp;                                /* The Registration Date and Time Time Stamp of files */
  Int                   size;                                     /* Data size (unit: bytes)                      */
  _ELIB_MS_DATE_DATA    ModifyDate;                               /* Renewal Date                                 */
  _ELIB_MS_DATE_DATA    CreateDate;                               /* Date of Creation                             */
  Int                   distribution_right;                       /* Redistribution Level Information             */
  Int                   audioCodecType;                           /* Audio Information                            */
  Int                   videoCodecType;                           /* Image Information                            */
  Int                   textType;                                 /* Text Information                             */
  Int                   fileType;                                 /* File Format                                  */
  unsigned long         live_info;                                /* Live Information (Because only ASF is used, do not use) */
  Int                   seek_info;                                /* Seek Possible Information                    */
  Int                   width;                                    /* Width (Image Size)                           */
  Int                   height;                                   /* Height (Image Size)                          */
  MSD_TIME              play_time;                                /* Replay Duration                              */
  Int                   audioBitRate;                             /* Audio Bit Rate                               */
  Int                   videoBitRate;                             /* Image Bit Rate                               */
  Char                  audioVendor[4];                           /* Audio Vendor Information                     */
  Char                  videoVendor[4];                           /* Image Vendor Information                     */
  _ELIB_MS_LIMITINFO    LimitInfo;                                /* Replay Restriction Information               */
  Int                   use_size;                                 /* File Use Size (unit: bytes) on FAT           */
  _ELIB_MS_SUBINFO      SubInfo;                                  /* Registration Contents SubInformation         */
                                                                  /* (such as UIM_ID)                             */
  unsigned short        LinkCount[ELIB_MS_LINK_MAX];              /* Link Information                             */
                                                                  /* File Name Character StringThe end is NULL.   */
  BYTE                  filename[ ELIB_MS_MNG_FILENAME_MAX ];     /* Title Character StringThe end is NULL.       */
                                                                  /* Title for management                         */
  BYTE                  title[ ELIB_MS_TITLE_MAX ];               /* Absolute Path Information                    */
  BYTE                  UrlPath[ELIB_MS_URLPATH_MAX];
  BYTE                  permission;                               /* File Access Authority                        */
  signed char           calling_link                              /* Incoming Image Setting Allowance Information */
  BYTE                  reserve[ 3 ];                             /* Reserve                                      */
}_ELIB_MS_TITLEINFO;
```

- Concerning File Access Authority
  The Return Value Define of the "File Access Authority Read" is used in the permission of the Title Information Structure.

- Redistribution Level Information
  The Return Value Define of the "C-7 Redistribution Level Information Acquisition" is used in the distribution_right of the Title Information Structure.
  * Refer to "Video Control Service Interface Specification".

- Incoming Image Setting Allowance Information
  The following define is used in the calling_link of the Title Information Structure.
  ELIB_MS_OK : Incoming Video Setting Allowed
  ELIB_MS_NG : Incoming Video Setting Not Allowed

- Concerning the File Format
  The Return Value Define of the "C-3 Format Acquisition" is used in the fileType of the Title Information Structure.
  * Refer to "Video Control Service Interface Specification".

- Concerning Audio Information
  The Remark Define of the "C-4 AV Information Acquisition" is used in the videoCodecType of the Title Information Structure.
  * Refer to "Video Control Service Interface Specification".

- Concerning Image Information
  The Remark Define of the "C-4 AV Information Acquisition" is used in the audioCodecType of the Title Information Structure.
  * Refer to "Video Control Service Interface Specification".

- Concerning Text Information
  The following define is used in the textType of the Title Information Structure.
  ELIB_MS_MEDIA_NOTEXIST : No Text Information
  ELIB_MS_MEDIA_T_QT : Text Information Available

· Concerning Audio Bit Rate
  If the audio codec is AAC, the value stored in the audioBitRate member is not valid.
  If the audio codec is AMR, variable length bit rate is possible but if it is variable length bit rate, the central value which is also the maximum value is set.

· Concerning Image Bit Rate
  The value of the image bit rate is not the actual value but a rounded value as shown below is set.
  up to 16kbps: 16kbps
  17 to 24kbps: 24kbps
  25 to 32kbps: 32kbps
  33 to 40kbps: 40kbps
  17 to 48kbps: 48kbps
  49 to 56kbps: 56kbps
  57 to 64kbps: 64kbps
  65 to 128kbps: 128kbps
  129 to 384kbps: 384kbps
  385 to 768kbps: 768kbps

· Concerning the sort key
  Acquire in the order of Registration Date: Conduct sort in the order of Terminal Time when the video data is registered in the FROM within the terminal.
  (* The order of the files may change according to the Clock Setting of the terminal.)
  Acquire in the order of Registration: Conduct sort in the order of when it is registered in the FROM within the terminal.

Acquire in the order of Title Name: Sort by the Title Name Character String (SJIS code).The overlapping items are distinguished by the Registration Order.
Acquire in the order of Date Created: Sort by the Date Created of the Title Information Structure.The overlapping items are distinguished by the Registration Order.
Acquire in the order of Data Size: Sort by the Data Size of the Title Information Structure.The overlapping items are distinguished by the Registration Order.

2.16 Acquisition of Free Space

Classification	Support Functions of the Video Control Service		
Function Name	Free Area Acquisition	Symbol	Elib_MS_GetFreeSpace

Functional Overview

*Get the free area (unit: bytes) of each designated folder (that can be used to save Video Data).
*The Maximum Free Area of the 22 folders "INBOX, Camera, and the User folders 1-20" are 3.1 MB in total.
*If free area is acquired by designating one of the 22 folders above, return the value calculated by subtracting the total size used by the 22 folders from 3.1 MB to AP as the free area.
*The free area returned with this function is the value calculated by subtracting the Video Data size and the Management File size from the folder capacity (Folder capacity - Video Data Size - Management File Size).

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_ GetFreeSpace (unsigned int Ap_ID, int Folder);		
Argument	Type	I/O	Description
Ap_ID	Unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
Return value	Type	-	Description
re t	Int	O	[Normal Completion] Free Area: Free Area (bytes) [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG
Remark			

The free area changes according to the Area Acquisition Unit of the file system.
For example, if File A= 12 bytes, File B= 15 bytes, and File C＝ 20 bytes,
if the file capacity is 100 bytes, the free area is 100-(12+15+20)= 58 bytes but if the Data Write Unit for the file system is 10 bytes, File A= 20 bytes, File B=20 bytes, and File C= 20 bytes,
therefore the free area is 100−(20+20+20)＝ 40 bytes.

2.17 Registered Movie Data Read

Classification	Support Functions of the Video Control Service		
Function Name	Registered Video Data Read	Symbol	Elib_MS_GetClip
Functional Overview			

Read the Video Data File (MP4) based on the designated folder identifier and the file number and set to the designated Read Buffer (secured on the AP side).
- It is possible to set how many bytes from the top of the file the Read File Data should begin reading.
- If the data of the desginated file largely exceeds the buffer compared with the Read Buffer secured in the application, the data is read up to the secured buffer and return ELIB_MS_NG_SIZESHORT (Buffer Inadequate) as a return value.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetClip(unsigned int Ap_ID, int Folder, int FileNum, int Offset, int Size, BYTE* Buffer);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Offset	Int	I	Offset from the top of the file (unit: bytes)
Size	Int	I	Size of secured buffer (unit: bytes)
Buffer	BYTE*	O	Pointer of buffer that writes the read data
Return value	Type	-	Description
re t	Int	O	[Normal Completion] Read Size : (byte) Buffer Inadequate : ELIB_MS_NG_SIZESHORT

			[Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Data Not Registered : ELIB_MS_DATANOT
Remark			
None			

2.18 Set Permitted Play Count

Classification	Support Functions of the Video Control Service		
Function Name	Replay Restriction Number Setting	Symbol	Elib_MS_SetPermittedPlayCount
Functional Overview			

- Set the number of times the Video registered in the Video of the designated folder was replayed.
- Information on the Replay Restriction Number is managed with the Replay Number Record up to now in the Movie File Management File.
- Return ELIB_MS_NG (Abnormal Completion) if the Number of times designation data of the Replay Restrictions during Video Data Registration is not registered.

Include file	Srv_ms_p.h
Calling Sequence	int Elib_MS_SetPermittedPlayCount (unsigned int Ap_ID, int Folder, int FileNum, int Count);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Count	Int	I	Number of times Replayed
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Data Not: ELIB_MS_DATANOT

			Registered

Remark
If the Number of times designation data of the Replay Restrictions is not designated during Video Data Registration, do not check (check to see if it exceeds the Maximum Number of times replayed) the Number of times the input value is replayed.（AP conducts check）

2.19 Set Nonrecognition of Permitted Play Count

Classification	Support Functions of the Video Control Service		
Function Name	Replay Restriction Not Allowed Setting	Symbol	Elib_MS_ClearPermission
Functional Overview			

- Set the Not Allowed Information in the Replay Not Allowed Information Record that corresponds to the file number of the designated folder.
- The Replay Restriction Not Allowed Information Record is managed by the Movie File Management File.
- The Replay Restriction Not Allowed Setting is an interface that sets information and the decision of whether or not to replay based on the setting is conducted by the AP.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_ClearPermission (unsigned int Ap_ID , int Folder, int FileNum);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Data Not Registered : ELIB_MS_DATANOT
Remark			

None

2.20 File Name Write

Classification	Support Functions of the Video Control Service		
Function Name	File Name Write	Symbol	Elib_MS_SetFilename
Functional Overview			

- Get the File Name Record on the Movie File Management File that corresponds to the file number of the designated folder.(The file name on FAT is not changed.)
- Return error if the video of the designated file number (FileNum) is not registered.
- The Maximum value of the File Name Character String is 60 bytes.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_SetFilename(unsigned int Ap_ID int Folder, int FileNum, BYTE* Filename);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Filename	BYTE*	I	The File Name Character String to be written (60 bytes or less and ends with NULL)
Return value	Type	I/O	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Data Not: ELIB_MS_DATANOT

			Registered
Remark			
None			

2.21 File Name Read

Classification	Support Functions of the Video Control Service		
Function Name	File Name Read	Symbol	Elib_MS_GetFilename
Functional Overview			

- Change the File Name Record on the Movie File Management File that corresponds to the file number of the designated folder.（(Do not refer to the file name on FAT.)
- Return error if the video of the designated file number (FileNum) is not registered.
- The Read File Name Character String Buffer is secured by AP.
- The Maximum size of the File Name Character String is 61 bytes.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetFilename (unsigned int Ap_ID, int Folder, int FileNum, BYTE* Filename);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Filename	BYTE*	O	Read File Name Character String Buffer Secure the buffer on the Caller side.
Return value	Type	I/O	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal : ELIB_MS_PARAM_NG

			Parameter Data Not Registered: ELIB_MS_DATANOT
Remark			
None			

2.22 Registered Movie Data Copy

Classification	Support Functions of the Video Control Service		
Function Name	Registered Video Data Copy	Symbol	Elib_MS_CopyClip
Functional Overview			

- Copy the file that corresponds to the file number in the designated folder.
- Assign the designated File Number to the copied file and manage by the individual Information Area (Record).
- Return error if the video of the designated file number (CFileNum) is not registered.
- If a Copy Desitination File Number exists (used) then return error.

Include file	srv_ms_p.h
Calling Sequence	int Elib_MS_CopyClip(unsigned int Ap_ID, int Folder, int CFileNum, int PFileNum);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
CFileNum	int	I	Copy Source File Number
PFileNum	int	I	Copy To File Number 0: Automatically assign the smallest unregistered file number. Other: Assign the designated number as the file number.

Return value	Type	I/O	Description
Ret	int	O	[Normal Completion] Copied File Number: 1- (positive value) - Maximum Video File Value: ELIB_MS_MP4FILE_MAX [Abnormal Completion]

			[Abnormal Completion]	: ELIB_MS_NG
			Abnormal Parameter	: ELIB_MS_PARAM_NG
			Data Not Registered	: ELIB_MS_DATANOT

Remark

- The copy of the Registered Video Data is copied within the same folder.(Copies cannot be made to other folders)

The file name of the FAT of the Copy Desitination File is given the default name at Video ELIB.

- Everything is also copied for the File Management Information in the Movie File Management File with the exception of the File Time Stamp.

2.23 File Access Authority Write

Classification	Support Functions of the Video Control Service		
Function Name	File Access Authority Write	Symbol	Elib_MS_SetFilePermission
Functional Overview			

- Set the File Access Authority managed by FAT that corresponds to the file number of the designated folder.
- Return error if the designated file number (FileNum) is not registered.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_SetFilePermission(unsigned int Ap_ID, int Folder, int FileNum, int Permission);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Permission	Int	I	File Access Authority ELIB_MS_FILE_RW: Read Write Allowed ELIB_MS_FILE_R: Read Only
Return value	Type	I/O	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Data Not: ELIB_MS_DATANOT

			Registered
Remark			
None			

2.24 File Access Permission

Classification	Support Functions of the Video Control Service		
Function Name	File Access Authority Read	Symbol	Elib_MS_GetFilePermission
Functional Overview			

- Get the File Access Authority managed by FAT that corresponds to the file number of the designated folder.
- Return error if the designated file number (FileNum) is not registered.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetFilePermission (unsigned int Ap_ID, int Folder, int FileNum);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Return value	Type	I/O	Description
Ret	Int	O	[Normal Completion] Read Write Allowed : ELIB_MS_FILE_RW Read Only : ELIB_MS_FILE_R [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Data Not Registered : ELIB_MS_DATANOT
Remark			

None

2.25 Movie Title Write

Classification	Support Functions of the Video Control Service		
Function Name	Video Title Write	Symbol	Elib_MS_SetTitle
Functional Overview			

- Change the management title of the file number in the designated folder.(The written title is not reflected in the header part of the Video Data Body.Only the Terminal Internal Information is rewritten.)
- If more than 256 bytes of data is input in the Title Character String, return an error.
- Return error if the designated file number (FileNum) is not registered.
- Change the Title for management but do not change the original title.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_SetTitle (unsigned int Ap_ID , int Folder, int FileNum, unsigned short Lang, BYTE* Title);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Lang	unsigned short	I	Write Title Language ELIB_MS_LANG_JPN: Japanese ELIB_MS_LANG_ENG: English ELIB_MS_LANG_MUL: Multilingual
Title	BYTE*	I	The Title Character String to be written (Ends with NULL)
Return value	Type	I/O	Description

ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Data Not Registered : ELIB_MS_DATANOT

Remark

The Write Title Character String should be 256 bytes or less and the character string should end with NULL.
The Argument Lang "Write Title Language" is ignored (Change the management title regardless of Language Setting).

2.26 Movie Title Read

Classification	Support Functions of the Video Control Service		
Function Name	Video Title Read	Symbol	Elib_MS_GetTitle
Functional Overview			

- Get the management title of the file number in the designated folder.(Refer to the Management Title Code in the individual Information Area prepared for file management and do not change the title in the Video File Header.))
- Return error if the designated file number (FileNum) is not registered.
- If the title of the management title does not exist, set NULL at the top of the Title Character string buffer.
- The original title can not be acquired.

Include file	srv_ms_p.h
Calling Sequence	int Elib_MS_GetTitle (unsigned int Ap_ID , int Folder, int FileNum, unsigned short Lang, BYTE* Title);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Lang	unsigned short	I	Read Title Language ELIB_MS_LANG_JPN: Japanese ELIB_MS_LANG_ENG: English ELIB_MS_LANG_MUL: Multilingual
Title	BYTE*	O	Read Title Character String Buffer Secure the buffer (257 bytes) on the Caller side.
Return value	Type	I/O	Description

ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Data Not Registered : ELIB_MS_DATANOT

Remark

- The Read Title Character String Buffer is secured on the application side.
The Argument Lang "Read Title Language" is ignored (Read the management title regardless of Language Setting).

2.27 Mail Attachment Movie Data Check Request

Classification	Support Functions of the Video Control Service		
Function Name	Mail Attachment Video Data Check Request	Symbol	Elib_MS_AttachChk

Functional Overview

- Check the video file that corresponds to the file number of the designated file identifier and return whether or not it can be attached to mail.
- Return error if the video of the designated file number (FileNum) is not registered.
- Check the following items for this function.
  (1)Is the video file an MP4 file?
  (2)Is the designated file size + the byte size of the URL character string of the video server + the byte size of the character string for the storage life more than 102400 bytes?
- Secure the area for the third argument sequence on the application side.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_AttachChk (unsigned int Ap_ID , int FileCount, _ELIB_MS_FILE_PATH *FilePatharray, int URLchara, int Timechara);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
FileCount	Int	I	Number of File Attachments
FilePatharray	_ELIB_MS_FILE_PATH *	I	The Folder Identifier of the File Attachment and the File Number Pointer Sequence
URLchara	Int	I	URL Character String of the Video Server (byte)
Timechara	Int	I	Number of characters in the Storage Life (byte)
Return value	Type	I/O	Description
ret	Int	O	[Normal Completion] Mail Attachment Allowed: ELIB_MS_OK Mail Attachment Not Allowed: ELIB_MS_NG [Abnormal Completion] Abnormal Parameter: ELIB_MS_PARAM_NG Data Not Registered: ELIB_MS_DATANOT

Remark

- The Folder Identifier of the file attachment and the structure of the File Number Sequence is as follows.

```
typedef struct tag_ELIB_MS_FILE_PATH {
   int Folder;              /*Folder Identifier*/
   int FileNum;             /*File Number*/
} _ELIB_MS_FILE_PATH
```

2.28 Acquisition Of Registered Movie Data Administration Information

Classification	Support Functions of the Video Control Service		
Function Name	Registered Video Data Information Acquisition	Symbol	Elib_MS_GetMovieInfo
Functional Overview			

- Get 1 item from the Title Information Structure that can be acquired with Registered Video Data List Acquisition using the File Number within the designated folder as a key.
- The provided data is management information on the individual information areas (record) created in the Video Data Registration.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetMovieInfo (unsigned int Ap_ID , int Folder, int FileNum, _ELIB_MS_TITLEINFO* Title);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	Int	I	File Number
Title	_ELIB_MS_TITLEINFO*	O	Title Information Structure
Return value	Type	I/O	Description
ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal : ELIB_MS_NG

			Completion] Abnormal Parameter : ELIB_MS_PARAM_NG Data Not Registered : ELIB_MS_DATANOT

Remark

- The area of the Title Information Structure is secured on the AP side.
- For the Title Information Structure, refer to Registered Video Data List Acquisition.

2.29 Cancel Movie Data Registration

Classification	Support Functions of the Video Control Service		
Function Name	Video Data Registration Cancellation	Symbol	Elib_MS_SetClipCancel

Functional Overview

- It corresponds with the interface of the B-2 Video Data Registration (asynchronous) , therefore cancel the registration process.
In response to the cancel request, return control to the caller and return the result as an event to the caller.
Return the result in the Data Registration Complete Event.
MSNotify_SET_CANCEL: Video Data Registration Cancellation Complete

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_SetClipCancel(unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

If the registration process is not executed (or if it is completed), return error (ELIB_MS_NG).
It is possible that the Data Registration Complete Event is returned after the Normal Completion is returned to the Cancel Request.
If the cancellation is complete, the Video Data File that is being created is deleted and it returns to the condition before registration.

2.30 Registered Movie Data Read (Asynchronous)

Classification	Support Functions of the Video Control Service		
Function Name	Registered Video Data Read (Asynchronous)	Symbol	Elib_MS_GetClipAsync
Functional Overview			

Read the Video Data File (MP4) of the file based on the designated folder identifier and the file number and set to the designated Read Buffer (secured on the AP side).
- It is possible to set how many bytes from the top of the file the Read File Data should begin reading.
- The read result is notified to the Caller by the event.
- If the data of the desginated file largely exceeds the buffer compared with the Read Buffer secured in the application, the data is read up to the secured buffer and notify ELIB_MS_NG_SIZESHORT (Buffer Inadequate) as detailed information of the event.
- The Registered Video Data Read (Asynchronous) (Simultaneous Multiple Read) of the Registered Video Data Read (Asynchronous) returns error.The Registered Video Data Read (Asynchronous) during Video Data Registration (Asynchronous) also returns error.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetClipAsync(unsigned int Ap_ID, int Folder, int FileNum, int Offset, int Size, BYTE* Buffer);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
FileNum	int	I	File Number
Offset	int	I	Offset from the top of the file (unit: bytes)
Size	int	I	Size of secured buffer (unit: bytes)
Buffer	BYTE*	O	Pointer of buffer that writes the read data
Return value	Type	-	Description
ret	int	O	[Normal Completion]

			Normal Completion	: ELIB_MS_OK
			[Abnormal Completion]	
			[Abnormal Completion]	: ELIB_MS_NG
			Abnormal Parameter	: ELIB_MS_PARAM_NG
			Data Not Registered	: ELIB_MS_DATANOT

Remark
The result is returned by the complete event (MSNotify_GET_DONE) and the read buffer size or if the buffer is inadequate then the error information is returned in the detailed information of the event.

2.31 Cancel Movie Data Read

Classification	Support function for video control service		
Function	Cancel movie data read	Symbol	Elib_MS_GetClipCancel

Functional overview

- It is for the interface that reads (asynchronous) the registered movie data, and cancels the reading processing.
- To the cancel request, it immediately returns control to the calling side and returns the result to the calling side as event.
- It returns the result by Cancel reading completed event.
  MSNotify_GET_CANCEL: Cancel reading movie data completed

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_GetClipCancel(unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
ret	Int	O	[At normal end] Normal end : ELIB_MS_OK [At abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- If the reading processing for the registered data has not been executed (including the case where it is already completed), it returns error (ELIB_MS_NG).
- Read data completed event might return after returning normal end to cancel request. (It never happens that both Read completed and Cancel read completed return.)

2.32 Change Downloading Date And Hour

Classification	Support Functions of the Video Control Service		
Function Name	Download Data and Time Modification	Symbol	Elib_MS_ChangeDownloadDate
Functional Overview			

Modify the Dowload Date and Time Information acquired from the mobile device of the Replay Restriction Information in the Registered Contents of all folders (INBOX, Camer, User Folder 1-20).
- Conduct modification process only with contents that have Replay Restriction and Replay is allowed.(The Modification Process is not conducted for contents that were once set as Replay Not Allowed.)
- Calculate the new and old Date and Time Information (unit: seconds) and modify the Download Date and Time Information (acquired from the mobile device).(Be sure to conduct leap year process on the caller side.)
- If the Date and Time of the mobile device before modification is not set, set the Date and Time before modification (Old) as 0.
  In such cases, make all contents with Replay Restrictions as Replay Not Allowed with the exception of the Restriction of the Number of times conducted.(Replay Not Allowed is not a normal setting so the "Not Allowed Setting Value by Download Date and Time Modification" is set at -1.)
- Ignore the value designated in the Argument Folder.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_ChangeDownloadDate(unsigned int Ap_ID, int Folder, iunsigned int New, unsigned int Old);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Folder	Int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
New	unsigned int	I	Representation in Seconds of the Date and Time after Modification (The number of seconds passed from the Standard Date and Time 1970/01/01)
Old	unsigned int	I	Representation in Seconds of the Date and Time before Modification (The number of seconds

			passed from the Standard Date and Time 1970/01/01) If the time setting of the mobile device is not conducted, set it at 0.
Return value	Type	-	Description
ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

- If the Download Date and Time after modification is not within the range supported by the Calendar API, 2002/01/01 thru 2099/12/31, then it is an error (ELIB_MS_PARAM_NG).
- the download time is not retained internally from the standard time in seconds, therefore a conversion API of the Calendar API is used to convert from Date and Time to seconds and from seconds to Date and Time.
- The specification is one that the downloaded video file can be moved between the INBOX folder, Camera folder and the User folders 1- 20, therefore the process is conducted not only on folders designated in Argument Folder but on all folders.

2.33 Acquisition Of Total Number Of Movie Data

Classification	Support Functions of the Video Control Service		
Function Name	Video Data Total Acquisition	Symbol	Elib_MS_GetMovieNum
Functional Overview			

- Get the total number of video files (data) within the designated folder.
- - If the total number of video files (data) is acquired, Initial Process does not need to be conducted.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetMovieNum (int Folder);		
Argument	Type	I/O	Description
Folder	int	I	Folder Identifier ELIB_MS_INB_FOLDER: INBOX ELIB_MS_CAM_FOLDER: Camera ELIB_MS_U01_FOLDER: User Folder 1 ELIB_MS_U02_FOLDER: User Folder 2 ELIB_MS_U03_FOLDER: User Folder 3 ELIB_MS_U04_FOLDER: User Folder 4 ELIB_MS_U05_FOLDER: User Folder 5 ELIB_MS_U06_FOLDER: User Folder 6 ELIB_MS_U07_FOLDER: User Folder 7 ELIB_MS_U08_FOLDER: User Folder 8 ELIB_MS_U09_FOLDER: User Folder 9 ELIB_MS_U10_FOLDER: User Folder 10 ELIB_MS_U11_FOLDER: User Folder 11 ELIB_MS_U12_FOLDER: User Folder 12 ELIB_MS_U13_FOLDER: User Folder 13 ELIB_MS_U14_FOLDER: User Folder 14 ELIB_MS_U15_FOLDER: User Folder 15 ELIB_MS_U16_FOLDER: User Folder 16 ELIB_MS_U17_FOLDER: User Folder 17 ELIB_MS_U18_FOLDER: User Folder 18 ELIB_MS_U19_FOLDER: User Folder 19 ELIB_MS_U20_FOLDER: User Folder 20
Return value	Type	-	Description
ret	Int	O	[Normal Completion] Normal Completion: 0 - Number of Video Files (data) [Abnormal Completion] Abnormal Completion: ELIB_MS_NG

Remark
None

2.34 Format Acquisition (For SD Menu Display)

Classification	Support Functions of the Video Control Service		
Function Name	Format Acquisition (Menu)	Symbol	Elib_MS_GetFormat_Menu
Functional Overview			

- Get the Video Format of the video information designated by the Storage Source Information.
If this File Format is acquired, Initial Process does not need to be conducted.
- The designation method for the Information Acquisition Source is the same as Video Information Registration (refer to Remarks).
This interface only provides FUI therefore as a rule the other functions are not used.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetFormat_Menu (_ELIB_MS_LOCATION *ClipLocation);		
Argument	Type	I/O	Description
ClipLocation	_ELIB_MS_LOCATION *	I	Data Acquisition Source Video Data (location)
Return value	Type	-	Description
ret	Int	O	[Normal Completion] MobileASF : ELIB_MS_MOBILEASF ASF : ELIB_MS_ASF MobileMP4 : ELIB_MS_MP4 MP4 : ELIB_MS_MP4 [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

If a file that does not belong to any file format such as MP4 or ASF is designated, return ELIB_MS_NG.
If the designation of the Information Acquisition Source is mistaken, return ELIB_MS_PARAM_NG.

2.35 Acquisition Of MP4 Data Size

Classification	Support function for video control service		
Function	MP4 data size acquisition	Symbol	Elib_MS_Get_MP4Size

Functional overview

- It parses MP4 data on memory for the size specified by the argument and gets MP4 data size.
- If MP4 data exists covering only some part, it gets the size obtainable by parsing that data.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_MP4Size (unsigned int Ap_ID, BYTE *Data, int *Size);		
Argument	**Type**	**I/O**	**Description**
Ap_ID	unsigned int	I	Application ID
Data	BYTE*	I	MP4Head address of data
Size	int*	I/O	MP4Data size(in terms ofbyte) It parsesMP4data for the size that has been input and returns the parse resultMP4size.
Return value	**Type**	**-**	**Description**
ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- If a file other than MP4 is specified, it returns ELIB_MS_NG.
- If the argument pointer is NULL, it returns ELIB_MS_PARAM_NG.
- If the data size value that is set as argument is less than the minimum executable size (30 bytes) for the header analysis processing, the format cannot be determined, therefore, even if the data is actually the format other than MP4, the analysis result will not be ELIB_MS_NG and the analysis result data size will be 0 byte instead, and then ELIB_MS_OK returns.

2.36 Acquisition of Movie ID

Classification	Support Functions of the Video Control Service		
Function Name	Video ID Acquisition Request	Symbol	Elib_MS_GetMovieID
Functional Overview			

- Get the Video ID that corresponds with the File Number/Folder Number.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetMovieID(unsigned int Ap_ID, int FileNum, int Folder, unsigned short *MovieID) ;		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
FileNum	int	I	File Number
Folder	int	I	Folder Number
MovieID	unsigned short *	I/O	Video ID
Return value	Type	-	Description
Ret	int	O	Normal Completion : ELIB_MS_OK [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

The video ID that this function returns has the following specification to avoid overlapping with the sound ID.
Bit15: Vidoe ID/Sound ID Identification Flag (1 fixed for Video ID)
Bit14-0: ID value (1- ELIB_MS_MP4FILE_MAX)
Note) The ELIB_MS_MP4FILE_MAX is 100 (64h).

2.37 Acquisition Of File Identifier/File Number(Movie ID)

Classification	Support Functions of the Video Control Service		
Function Name	File Number Acquisition Request	Symbol	Elib_MS_GetFolderFileNum
Functional Overview			

- Get the File Number/Folder Number that corresponds with the Video ID.

/* File Number/Folder Number Storage Sequence */
FileInfo[0] --- File Number
FileInfo[1] --- Folder Number

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetFolderFileNum(unsigned int Ap_ID, unsigned short MovieID, int *FileInfo) ;		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
MovieID	unsigned short	I	Video ID
FileInfo	int *	I/O	Pointer of the Sequence that stores the file number/folder number
Return value	Type	-	Description
ret	int	O	Normal Completion : ELIB_MS_OK [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

The Sequence that stores the file number/folder number should be secured for two elements.

2.38 Movie Information Registration

Classification	Support Functions of the Video Control Service		
Function Name	Video Information Registration	Symbol	Elib_MS_SetClipInfo
Functional Overview			

Acquire and register the various information of the Video Data in the designated memory or file format into the temporary area.
If the function has Normal Complete, the "Video Information Number", the key to accessing the registered information is acquired.

In order to conduct Information Acquisition/Setting Process of Group C, this function must first be executed.
If the information is accessed with the Group C function, use the Video Information Number acquired with this function.

This function must be executed even when conducting Replay Operation of Group D and Edit Operation of Group F.(The Replay target designation is "Video Information Number".)

When conducting Telephone Incoming Video Replay, set ELIB_MS_LOC_ID_CALL_MOV as the Location Type of the ClipLocation.
ELIB_MS_LOC_ID_CALL_MOV is only valid in this API.(Its use in other APIs is forbidden.)

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_SetClipInfo (unsigned int Ap_ID, _ELIB_MS_LOCATION *ClipLocation);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipLocation	_ELIB_MS_LOCATION *	I	Data Acquisition Source Video Data (location)
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] Registered Video Information Number (ClipID) : 1～ [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Size Inadequate : ELIB_MS_NG_SIZESHORT Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

For Registration Requests that exceed the Number of Video Information that can be registered (ELIB_CLIPINFO_MAX), return ELIB_MS_NG.

- If the data that is being received is not enough for Information Acquisition, return ELIB_MS_NG_SIZESHORT.If there is still a possibility of reception, the Video Information Registration can be retried.(Process during Pseudo Streaming

- The information that is laid out in the temporary area is as follows.
  (1) Video Data Format (such as ASF/MP4)
  (2) Date Information (Date Renewed, Date Created)
  (3) Redistribution Allowed Information

(4) Seek Allowed/Not Allowed Information
(5) Replay Description Information Structure (Information such as Title, Copyright, Description in Japanese, English, Multilingual is laid out.)
(6) Image Size (Height, Width)
(7) Codec Information (Audio, Video, Text)
(8) Replay Duration
(9) Incoming Image Setting Allowance Information

All the information above can be read from the function of Group C.

[Information Type and Group C API Correspondence Table]

Table Information Type and Group C API Correspondence Table

	Information Type	Corresponding Group C API
1	Video Data Format (such as ASF/MP4)	C-3 Format Acquisition
2	Date Information (Date Renewed, Date Created)	C-21 Time Information Acquisition in Contents
3	Redistribution Allowed Information	C-7 Redistribution Level Information Acquisition
4	Seek Allowed/Not Allowed Information	C-15 Seek Information Acquisition
5	Replay Description Information Structure (Information such as Title, Copyright, Description in Japanese, English, Multilingual is laid out.)	C-6 Description Information Acquisition
6	Image Size (Height, Width)	C-5 Image Size Acquisition
7	Codec Information (Audio, Video, Text)	C-4 AV Information Acquisition
8	Replay Duration	C-17 Replay Total Time Acquisition
9	Incoming Image Setting Allowance Information	C-27 Incoming Image Setting Allowance Information

[Designation Method of the Registration Source Data]
(1) Designate the Location Type of the Input Parameter ClipLocation.

ELIB_MS_LOC_MEM : Continuous Memory (CPU Address Space)
ELIB_MS_LOC_ID : File that is registered in Flash (File Number)
ELIB_MS_LOC_PATH : Direct Designation of file using full path (when accessing SD)
ELIB_MS_LOC_BUFF : Discontinuous Memory: Buffer (CPU Address Space)
ELIB_MS_LOC_ID_CALL_MOV: ELIB_MS_LOC_ID for Telephone Incoming Video Replay (only valid in this API)

(2) Designate a Location for ClipLocation that goes with the LocationType.
In the case of Continuous Memory Designation
: Set the Top Address and the size presently acquired and the overall size of the memory in the LocMem.
Set "MaxSize: Total size" for the data size and the value set in "Size: size" can be incomplete data.
(The Mid-Reception Data during Pseudo Streaming can also be registered.)

In the case of File Number designation
: Designate the Folder Number and the File Number..
(Designate the following Folder identifier as the Folder Number.)

Folder Identifier
ELIB_MS_INB_FOLDER: INBOX
ELIB_MS_CAM_FOLDER: Camera
ELIB_MS_U01_FOLDER: User Folder 1
ELIB_MS_U02_FOLDER: User Folder 2
ELIB_MS_U03_FOLDER: User Folder 3
ELIB_MS_U04_FOLDER: User Folder 4
ELIB_MS_U05_FOLDER: User Folder 5
ELIB_MS_U06_FOLDER: User Folder 6
ELIB_MS_U07_FOLDER: User Folder 7

ELIB_MS_U08_FOLDER: User Folder 8
ELIB_MS_U09_FOLDER: User Folder 9
ELIB_MS_U10_FOLDER: User Folder 10
ELIB_MS_U011_FOLDER: User Folder 11
ELIB_MS_U12_FOLDER: User Folder 12
ELIB_MS_U13_FOLDER: User Folder 13
ELIB_MS_U14_FOLDER: User Folder 14
ELIB_MS_U15_FOLDER: User Folder 15
ELIB_MS_U16_FOLDER: User Folder 16
ELIB_MS_U017_FOLDER: User Folder 17
ELIB_MS_U018_FOLDER: User Folder 18
ELIB_MS_U19_FOLDER: User Folder 19
ELIB_MS_U20_FOLDER: User Folder 20

In the case of File Name designation
: Designate the file name with full path.Be sure the file name ends with "NULL".

In the case of Discontinuous Memory Designation
: Set the Top Address and size of the data received in one time and the cumulative total value and overall size of those that are already acquired.
The designated data is the information in A to A (A<B) of all the data.
When starting Streaming Replay, make sure the size set is less than 160 KB.
If more than 160 KB is set, the Replay Operation is not guaranteed.
If this memory is designated, from the start of Data Reception to the Video Registration Completion (Max 120 KB) should be in the Continuous Memory.
(In order to make the Video Information Registration Process possible when streaming.)
If this location is designated, the basic operations during Registration are the same as when ELIB_MS_LOC_MEM is designated.
If the Replay Process is conducted after Registration is complete, it is necessary to conduct Release Control of the designated memory.

In the case of File Number designation for Telephone Incoming Video Replay
: Designate the Folder Number and the File Number..
Be sure to set in the LocID member for the File Number Designation.It does not have a member for Telephone Incoming Video Replay.

```
typedef struct tag_ELIB_MS_LOC_MEM {
    BYTE        *Data;          /* Data Address                          */
    int         Size;           /* Partial Data Size (Pseudo-Streaming)  */
    int         MaxSize;        /* All Data Size                         */
} _ELIB_MS_LOC_MEM;

typedef struct tag_ELIB_MS_LOC_ID {
    int         Folder;         /* Folder Identifier (Number)            */
    int         FileID;         /* File Number                           */
} _ELIB_MS_LOC_ID;


typedef struct tag_ELIB_MS_LOC_PATH {
    BYTE        *FilePath;      /* Top address of the File Path Character String */
} _ELIB_MS_LOC_PATH;


typedef struct tag_ELIB_MS_LOC_BUFF {
    BYTE        *Data;          /* The Data Address received this time   */
    int         RcvSize;        /* The Data size received this time      */
```

```
    int         Size;           /* Cumulative Size of Received Data          */
    int         MaxSize;        /* All Data Size to be received              */
} _ELIB_MS_LOC_BUFF;

typedef struct tag_ELIB_MS_LOCATION {
    Int                         LocationType;   /* Data Registration Source Type                 */
    union {
        _ELIB_MS_LOC_MEM LocMem;        /* (Continuous) Memory Designation               */
        _ELIB_MS_LOC_ID     LocID;      /* File ID Designation                           */
        _ELIB_MS_LOC_PATHLocPath;       /* Full Path Designation                         */
        _ELIB_MS_LOC_BUFFLocBuff;       /* Discontinuous Memory Designation              */
    } Location;                         /* Information for each Registration Source Type*/
} _ELIB_MS_LOCATION;
```

Set the top address of the received data in the _ELIB_MS_LOC_BUFF Data.Set it so that the RcvSize shows the received data size, the Size shows the Cumulative size of the received data and the MazSize shows the expected received data size when all reception is complete.RcvSize = Size during Video Information Registration and in retry when the reception data is insufficient, do not change the location of the data.(The Additionally Received Data is stored in the continuous memory and the previous received data is combined to make the input parameter for Video Information Registration Execution.）

2.39 Erase Movie Information

Classification	Support function for video control service		
Function	Erase movie information	Symbol	Elib_MS_DelClipInfo
Functional overview			

Erase the specified movie information that has been registered.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_DelClipInfo (unsigned int Ap_ID , int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number Delete one : Set video information number Delete all : ELIB_MS_NUMBER_ALL
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- If the video information number of a movie being on play is specified as ClipID, and if Delete All is specified while a movie is playing, the API executes abnormal end.

2.40 Format Acquisition

Classification	Support function for video control service		
Function	Get format	Symbol	Elib_MS_GetFormat
Functional overview			

Get the format for movie data which information is already registered.
Get information on the file format (ASF/MP4).

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_GetFormat(unsigned int Ap_ID , int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
Return value	Type	-	Description
Ret	int	O	[Normal end] MobileASF : ELIB_MS_MOBILEASF ASF : ELIB_MS_ASF MobileMP4 : ELIB_MS_MOBILEMP4 MP4 : ELIB_MS_MP4 [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Unregistered video information : ELIB_MS_DATANOT

Remark

- If the format of movie data is unidentifiable, it returns ELIB_MS_NG.
  (ASF/MP4 that is not playable is also regarded as unidentifiable movie data.)
- Since this API gets file format by parsing the header of movie data, file access including semaphore and file open/close occurs and the processing is delayed.

2.41 AV Information Acquisition

Classification	Support function for video control service		
Function	Get AVInformation	Symbol	Elib_MS_GetAVInfo
Functional overview			

For movie data which information is already registered, get information for existence or nonexistence of respective streams for video/audio/text and for the codec.

Include file	srv_ms_p.h
Calling sequence	Elib_MS_GetAVInfo(unsigned int Ap_ID, int ClipID, int* audioCodecType, int* videoCodecType int* TextCodecType);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
audioCodecType	int*	O	Audio codec
videoCodecType	int*	O	Video codec
TextCodecType	int*	O	Text-related (QuickTimeText etc.)

Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- If stream exists, the following value is set.

(1) audioCodecType:
ELIB_MS_MEDIA_A_AMR
ELIB_MS_MEDIA_A_G726
ELIB_MS_MEDIA_A_AAC

(2) videoCodecType:
ELIB_MS_MEDIA_V_MPG4
ELIB_MS_MEDIA_V_H263
ELIB_MS_MEDIA_V_H263BL

(3) textCodecType:

ELIB_MS_MEDIA_T_QT

- If stream does not exist, ELIB_MS_MEDIA_NOTEXIST is set to the codec type of audio/video/text.
- For unsupported codec, ELIB_MS_MEDIA_NOTSUPPORT is set.
- In the case of being judged as non-compliant media in each media by media type (AVT), set ELIB_MS_MEDIA_NOTSUPPORT.

(*)H263 ShortHaeder and H263 baseline are different only in the way of storing within MP4 and both of them are H263 as codec.

2.42 Acquisition Of Video Picture Size

Classification	Support function for video control service		
Function	Get Video Picture Size	Symbol	Elib_MS_GetFrameSize

Functional overview

Get the video picture size included in the movie data which information is already registered.
Video picture size applies the value that is registered as header information of movie data and is not the result obtained by actually decoding video data.

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_GetFrameSize (unsigned int Ap_ID, int ClipID, int* Width, int* Height);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Width	int*	O	Video picture width
Height	int*	O	Video picture height

Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- Information acquirable by this API is the information that is obtained from the header of multiplex file and may differ with actual movie data size. The actual movie data size is included in The front picture play completed event.
  How to leverage both data is appli-dependent.

- If there is no video data, it returns ELIB_MS_OK and 0 is set to both Width and Height.

2.43 Acquisition of Descriptive Information

Classification	Support function for video control service		
Function	Get Descriptive Information	Symbol	Elib_MS_GetDescription
Functional overview			

Get information (mostly string) describing the movie data included in the movie data which information is already registered.

Include file	Srv_ms_p.h		
Calling sequence	int Elib_MS_GetDescription (unsigned int Ap_ID, int ClipID, unsigned short Lang, _ELIB_MS_DESCRIPTION *DataInfo);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Lang	unsigned short	I	Title language to read ELIB_MS_LANG_JPN : Japanese ELIB_MS_LANG_ENG : English ELIB_MS_LANG_MUL : Multilingual ELIB_MS_LANG_TITLE: Management title
DataInfo	_ELIB_MS_DESCRIPTION *	O	Moving image information structure storage pointer
Return value	Type	I/O	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

The following area should be freed up by the calling side.
Moving image information structure

```
Typedef struct tag_ELIB_MS_DESCRIPTION{
        BYTE   title[ELIB_MS_TITLE_MAX];          //Title
        BYTE   author[ELIB_MS_AUTH_MAX];          //Author
        BYTE   copyright[ELIB_MS_CPY_R_MAX];   //Copyright (copyright)
        BYTE   description[ELIB_MS_DSCRP_MAX];//Description
} _ELIB_MS_DESCRIPTION;
```

*The maximum acquirable string length with each member is 257bytes of 256bytes string+terminating NULL.
*If management title is assigned to the argumentLang, the management title is set only fortitle, and forauthor, copyright, description, the value will be unset.
*If Japanese, English, multilingual is assigned to the argumentLang, the respective strings for title, author, copyright, descriptionwill be extracted from within the data ofASF/MP4.
*If you specify the video information registered by other thanELIB_MS_LOC_IDtoLocationTypeof [C-1Video Information Registration] and specify management title to the argumentLang, it returns

Abnormal end (ELIB_MS_NG).
*For character code(UCS-2/UTF-8)which is dependent onASF/MP4, it is converted toSJIS.
*If there is no corresponding data, set [NULL] to the head.

2.44 Acquisition of QTT Header Information (MP4)

Classification	Support function for video control service		
Function	Get QTT Header Information (MP4)	Symbol	Elib_MS_Get_QTTHeader
Functional overview			

Get QuickTimeText header information included in the movie data which information is already registered.
MP4may include text stream in addition to image/audio. This text data is called QuickTimeText.
QuickTimeText has the parameters set in the header by which to display what you want, and you can get the information from this function.

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_Get_QTTHeader(unsigned int Ap_ID , int ClipID, int* Tx, int* Ty, int* TrackWidth, int* TrackHeight, int* NumOfTextHeader, _ELIB_MS_MP4_TEXT_INFO * MP4TextInfo);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
Tx	int*	O	TextTrackhorizontal alignment(pixel)
Ty	int*	O	TextTrackvertical alignment(pixel)
TrackWidth	int*	O	TextTrackwidth(pixel)
TrackHeight	int*	O	TextTrackheight(pixel)
NumOfTextHeader	int*	O	ELIB_MS_MP4_TEXT_INFOnumber(Max24th at follows)
MP4TextInfo	_ELIB_MS_MP4_TEXT_INFO *	O	Head address of QuickTimeTextheader information storage area Calling side should free up the area (structure array) for the number of maximumMaxInfoNum=24and set the head address. It becomesIndexnumber in presence order and ascending order.

Return value	Type	I/O	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark	

- How to reflect the header file content acquired from this function in the display is application-dependent.

MaxInfoNum=24 ;

QTT header information structure: 24byte

```
typedef struct tag_ELIB_MS_ MP4_TEXT_INFO {
  unsigned int    DisplayFlags;            //Flag that shows text operation etc.(*1)
  signed char     HorizontalJustification; //Shows horizontal alignment position of text(signed int of 1 byte)(*2)
  signed char     VerticalJustification;   //Shows vertical alignment position of text(signed int of 1 byte)(*2)
  unsigned char   BackgroundColor_R;       //Shows red color component of text box background color.
  unsigned char   BackgroundColor_G;       //Shows green color component of text box background color.
  unsigned char   BackgroundColor_B;       //Shows blue color component of text box background color.
  unsigned char   BackgroundColor_A;       //Shows transparency component of text box background color.
  Short           DefaultTextBox_top;      //Shows the top position of text box.
  Short           DefaultTextBox_left;     //Shows the extreme left position of text box.
  Short           DefaultTextBox_bottom;   //Shows the bottom+1 position of text box.
  Short           DefaultTextBox_right;    //Shows the extreme right+1 position of text box.
  unsigned char   FaceStyleFlag;           //Character decoration specification (such as underline)
  unsigned char   TextColor_R;             //Shows red color component of text color.
  unsigned char   TextColor_G;             //Shows green color component of text color.
  unsigned char   TextColor_B;             //Shows blue color component of text color.
  unsigned char   FontSize;                //Font size(12 dots or 24 dots)
  unsigned char   reserve[1];              //Boundary
} _ELIB_MS_MP4_TEXT_INFO
```

- If text stream is not included in MP4 or in the case of ASF, it makes header count=0 and does not generate an error.
- Field value in TextBox is converted to LittleEndian in the movie Elib.
- For details on the data of Tx,Ty,TextWidth and TextHeight, see 3.1.3.1. Track Header Atom of MP4Specification for Mobile Terminals.
- For details on the data of QTT header information structure, see 3.1.3.3.3.3.8. TextSampleEntry of MP4Specification for Mobile Terminals.
- For font size, 12 dots and 24 dots are selectable. (The set value is 12 or 24)

*1) For details on DisplayFlags, there is the description in 3GPP specification (http://www.3gpp.org/). The excerpt is as follows.

DisplayFlags

0x00000020 Scroll In
0x00000040 Scroll Out
0x00000180 Scroll Direction
0x00000800 continuous karaoke
0x00020000 write text vertically

Scroll Direction in DisplayFlags

00b – Vertical scroll from bottom to top ('credits style')
01b – Horizontal scroll from right to left ('marquee style')
10b – Vertical scroll from up to bottom
11b – Horizontal scroll from left to right

*2) For HorizontalJustification/VerticalJustification, there is the description in 3GPP specification (http://www.3gpp.org/). The excerpt is as follows.

HorizontalJustification

0: left justification, 1: center justification, -1: right justification

VerticalJustification

0: top justification, 1: center justification, -1: bottom justification

2.45 Acquisition Of Movie Audio Volume

Classification	Support function for video control service		
Function	Get movie audio volume	Symbol	Elib_MS_GetVolume

Functional overview

Get speaker audio volume of movie audio.
Movie audio volume is not a value that is set for each movie data. Therefore, the movie audio volume to be obtained from this function is the one reflected in all the movie playings.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_GetVolume (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Set volume level 0 : Silent 1 to 6 : Volume level 1 to 6 [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

Nothing in particular.

2.46 Movie Audio Volume Setting

Classification	Support function for video control service		
Function	Set movie audio volume	Symbol	Elib_MS_SetVolume

Functional overview

Set speaker audio volume of movie audio.
Movie audio volume is not a value that is set for each movie data. Therefore, the movie sound volume to be set by this function will be reflected in all the movie playings.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_SetVolume (unsigned int Ap_ID, int Volume);		
Argument	**Type**	**I/O**	**Description**
Ap_ID	unsigned int	I	Application ID
Volume	int	I	Set volume level 0 : Silent 1 to 6 : Volume level 1 to6
Return value	**Type**	**I/O**	**Description**
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- Movie audio volume basically interfaces with ring alert. This function is required for changing the movie audio volume to the volume level different from the ring alert while the movie is playing.
- Movie audio volume reaches the same volume level as the ring alert when the movie starts to play. If the audio volume is set in advance and then the movie starts to play, the audio volume will not be reflected.

2.47 Acquisition Of Play Mode

Classification	Support function for video control service		
Function	Get play mode	Symbol	Elib_MS_GetPlayMode

Functional overview

Get the play mode of video control service (on play, on recording, on registration, on editing, on play+recording, on play+registration, unprocessed).

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_GetPlayMode (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Play status Movie is on play. : ELIB_MS_PLAY_ON Movie is on recording: ELIB_MS_REC_ON Movie is on registration: ELIB_MS_SET_ON Movie is on editing: ELIB_MS_EDIT_ON Movie is on: play+recording ELIB_MS_PLAY_REC_ON Movie is on: play+registration ELIB_MS_PLAY_SET_ON Unprocessed : ELIB_MS_IDLE [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

Nothing in particular

2.48 Acquisition Of MP4 File Type Information

Classification	Support function for video control service		
Function	Get MP4 file type Information	Symbol	Elib_MS_Get_FileType
Functional overview			

Determine whether or not the the movie data existing in the specified address is available for pseudo-streaming.
This function should be executed only for the one that Application can recognize as movie data=MP4 to be covered.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_FileType(unsigned int Ap_ID , BYTE* DounloadData, int* DataSize);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
DownloadData	BYTE*	I	Head address of movie data
DataSize	int*	I/O	Movie data size(in terms ofbyte) When the return value is [Short of size], the function returns the number of bytes(SizevalueofftypAtom+16)necessary for analysis. (*)16bytes is for determining the nextAtomor the size
Return value	Type	I/O	Description
Ret	int	O	Streaming available : ELIB_MS_SPOK Pseudo-streaming available : ELIB_MS_OK Pseudo-streaming unavailable & DL play available : ELIB_MS_DPNG_DLOK DL play available: (user confirmation required) ELIB_MS_DLOK_UC DL unavailable : ELIB_MS_DLNG Short of size : ELIB_MS_NG_SIZESHORT Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- Not all of the movie data for passing to this function should be prepared.

- Information to be given by this function will be the same as the identification of MobileMP4 and ordinary MP4 for the acquisition of format obtainable through registration of movie data, but since only small portion of data is checked, the consistency of other movie data is ignored.
  To determine whether or not it is playable, execute the video information registration processing and

obtain the result.

- This function is mainly used for promptly determining whether or not the movie data that you started to download is available for pseudo-streaming.
  Application should check whether or not the movie data is available for pseudo-streaming by using this function in the stage when it gets the download data of the movie data.
  If the downloaded data is judged as insufficient for determination (ELIB_MS_NG_SIZESHORT), it should make the judgement by executing again this function in the stage when the downloaded data is added.

- Content of check for this function is as follows.
(1) If FileTypeAtom exists at the head of the data and the content is Major-brand="mmp4", and either of CompatibleBrand is "mmp4", pseudo-streaming will be available. Moreover, if the next Atom is moov and the size of ftyp plus moov is within the range of streaming playable (120KB), it will be available for streaming(*).
(2) For the case other than (1), if FileTypeAtom exists at the head of the data and the content is Major-brand="mmp4" or "3gp4" or "3gp5", or either of CompatibleBrand is "mmp4" or "3gp4" or "3gp5", it will be unavailable for pseudo-streaming & available for DL play.
(3) For the case other than (1)(2), if FileTypeAtom exists at the head of the data and the content is Major-brand="mp41" or "mp42" or "isom", or either of CompatibleBrand is "mp41" or "mp42" or "isom", it will be available for DL play (user confirmation required).
(4) For cases other than the above (specified brand does not exist, FileTypeAtom does not exist, etc.), DL will be unperformable.

(*) That is to say, the data that is determined to be available for streaming is also available for pseudo-streaming.

- When the return value of this function is short of size, the function returns the number of bytes necessary for analysis by DataSize.

- It is impossible for this function to determine whether or not streaming is performable. However, use this function when conducting streaming, check for [Streaming performable] and conduct the streaming play of the file [Streaming performable]. (After the file starts to play, judgement of whether or not streaming is performable is made as needed, and if noncompliance is detected, you can receive [Play error] during the playing processing.
  As error during the playing
    ELIB_MS_PERR_STMHSIZE: Noncompliance with header size during streaming
    ELIB_MS_PERR_STMFMT : Noncompliance with MP4 file format during streaming
    ELIB_MS_PERR_STMBUF : Noncompliance with buffer model during streaming
exist. (Other than those above, noncompliance occurs for the existing errors.)
(*) By adding the judgement "Streaming performable", it is not notified practically.

2.49 Movie Title Write (Video Information Number)

Classification	Support function for video control service		
Function	Movie Title Writing (Video Information Number)	Symbol	Elib_MS_SetTitle_Clip

Functional overview

- Change the title registered in the temporary area for the specified video information number.
  (If the entity of video information number is the movie data on memory, change the title of header also. If the entity is file, change the title string for administrating the movie file management file instead of changing the title of header. For the movie data on memory, if the title information of the specified language does not exist in the header, it will generate an error. : ELIB_MS_DATANOT)
- Either could be selected for the input parameter [Lang] in case that the entity is file, but it will be ignored when changing the title of the movie file management file and the specified string will be input as it is.
- Writable title characters (byte) should be obtained from C-15 Acquisition Of The Number Of Movie Title Characters To Be Written and the number of characters equal or more than that should not be written.
- If the specified video information number is not registered, it returns error.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_SetTitle_Clip(unsigned int Ap_ID , int ClipID, unsiged short Lang, BYTE *Titile);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
Lang	unsigned short	I	Title language for writing ELIB_MS_LANG_JPN : Japanese ELIB_MS_LANG_ENG: English ELIB_MS_LANG_MUL: Multilingual
Title	BYTE*	I	Title string to be written (NULL termination)
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT
Remark			

2.50 Seek Information Acquisition

Classification	Support function for video control service		
Function	Get seek information	Symbol	Elib_MS_Get_Seek_Info
Functional overview			

- Get the the information about whether or not the movie data which information was already registered is available for seek.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_Seek_Info (unsigned int Ap_ID , int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Available for seek : ELIB_MS_OK Unavailable for seek: ELIB_MS_NG [Abnormal end] Parameter abnormal: ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT
Remark			

- Nothing in particular.

2.51 Acquisition Of The Number Of Movie Title Characters To Be Written

Classification	Support function for video control service		
Function	Get the number of movie title characaters to be written	Symbol	Elib_MS_GetTitle_Num

Functional overview

- This function is called when you change the title name of the movie data existing on memory by Movie Title Write (Video Information Number).
- In case that you change the title of the movie data on memory, you cannot change the title with the number of characters equal to or more than that of the title written in the current header because the title information existing in the header information of movie data is rewritten. Therefore it returns the number of bytes that allows you to change the title by this function.
- If the specified video information number is not registered, it returns error.
- For the movie data on memory, you can get the number of bytes writable for the title of the specified language. In case that the title for the specified language does not exist. "0" is set to return value.
- If the movie data existing on file is specified, it returns the maximum string length (256) of the title string for administration registered in the temporary area.
- For the movie data existing on file, input parameter [Lang] is ignored irrespective of either one is selected.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_GetTitle_Num(unsigned int Ap_ID , int ClipID unsigned short Lang);		
Argument	**Type**	**I/O**	**Description**
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
Lang	unsigned short	I	Title language for writing ELIB_MS_LANG_JPN : Japanese ELIB_MS_LANG_ENG: English ELIB_MS_LANG_MUL: Multilingual
Return value	**Type**	**I/O**	**Description**
Ret	int	O	[Normal end] The number of bytes for the title that can be changed 0 to [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- AP calls this function, grasps the number of bytes for the title that can be changed and imposes restrictions on the number of characters when changing the title.

2.52 Acquisition Of Total Play Time

Classification	Support function for video control service		
Function	Get total play time	Symbol	Elib_MS_Get_Play_Time
Functional overview			

- Get the total play time of the movie data of the specified video information number.
- If the specified video information number is not registered, it returns error.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_Play_Time(unsigned int Ap_ID , int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Total play time 0 to (unit: msec) [Abnormal end] Abnormal end : ELIB_MS_NG Unregistered data: ELIB_MS_DATANOT
Remark			

Nothing in particular

2.53 Acquisition of QTT Data Analysis Result Output Memory Size

Classification	Support function for video control service		
Function	Get QTTdataanalysis result outputmemory size	Symbol	Elib_MS_Get_QTTDataSize
Functional overview			

- Get memory size that outputs QTT data analysis result. Application should get memory for this size and conduct DTT data analysis.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_QTTDataSize(unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Memory size for QTTresult output [Abnormal end] ELIB_MS_NG
Remark			

Nothing in particular

2.54 QTT Data Analysis

Classification	Support function for video control service		
Function	QTT data analysis	Symbol	Elib_MS_Parse_QTTData
Functional overview			

- Parse QTT data and output the result to the specified memory.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Parse_QTTData (unsigned int Ap_ID , void* Address, unsigned int Size, _ELIB_MS_QTT_DATA * QTTData		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Address	void *	I	Head address of QTT data
Size	unsigned int	I	Memory size for QTTresult output
QTTData	_ELIB_MS_QTT_DATA*	I/O	Head address of the memory to which QTT analysis result is output
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- For QTT data head address of input parameter, set the head address of the data of the message received from the movie Elib.
- Application will display the text on LCD based on this data and the basic effect obtained by Elib_MS_Get_QTTHeader.
- MSNotify_PLAY_QTTDATA(QuickTimeText data) event may automatically eliminate event by the video control ELIB in the case where the release of Text data (QTT data) is not done in time by AP.
- Add final TEXT identification information to the analysis result.

2.55 Acquisition Of Maximum Movie Recording Data Size

Classification	Support function for video control service		
Function	Get maximum movie recording data size	Symbol	Elib_MS_Get_Max_MovieRec_Data_Size

Functional overview

- Return the maximum size of recordable movie data.
- However, it is not always Disk space left=Maximum size of recordable data, therefore, when you record [movie], get the maximum size from this function and do not set the value exceeding that maximum size to the maximum size in recording.

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_Get_Max_MovieRec_Data_Size (unsigned int Ap_ID , int StrageMedia, int RecodeFormat, unsigned int DiscFreeSpace);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
StrageMedia	int	I	Recording media ELIB_MS_SD : SD card ELIB_MS_FLASH: Internal FLASH
RecodeFormat	int	I	Recording format ELIB_MS_MP4 : MP4 ELIB_MS_MP4ST: MP4 available for pseudo-streaming ELIB_MS_ASF : ASF
DiscFreeSpace	unsigned int	I	Space left in SD/Internal FLASH (byte)

Return value	Type	I/O	Description
Ret	int	O	[Normal end] Maximum recordable size (byte) [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- It performs the following processing by the recording media and returns the value.

Case1 : SDCard(MP4)
Ret = DiscFreeSpace * α; // α is the ratio excluding the area for the managed information. The value is dependent on MP4 recording performance ofMSD (*1).

Case2 : Internal memory
Ret = DiscFreeSpace; // Internal memory is equivalent to the free space because it provides the working storage area for the management information.

Case3 : SDCard(ASF)
Ret = DiscFreeSpace * β; // β is the ratio excluding the area for the management information. The value is dependent on ASF recording performance ofMSD (*2).

(*1) The Current management information covers 50% of recording data size, which makes α=2/3.
(*2) The current management information covers 0% of recording data size, which makes β=1.

- The calculated value for the case where MP4 available for pseudo-streaming is specified restructures the media data header portion, therefore it will be as twice as large as the recording size.

2.56 Acquisition of Time Information In The Content

Classification	Support function for video control service		
Function	Get time information in the content	Symbol	Elib_MS_Get_Time_Info
Functional overview			

- Get [Created date and time] [Modified date and time] in the content (MP4, ASF) for the movie data which information is already registered.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_Time_Info(unsigned int Ap_ID, int ClipID, _ELIB_MS_DATE_DATA * CreateDate, _ELIB_MS_DATE_DATA * ModifyDate);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
CreateDate	_ELIB_MS_DATE_DATA *	O	Created date
ModifyDate	_ELIB_MS_DATE_DATA *	O	Modified date
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- For _ELIB_MS_DATE_DAT, see [Movie Data Registration].
- For the content (ASF) in which modified date and time does not exist, the modified date and time data containing NULL is returned.

2.57 Judgement Of Whether Or Not To Cut Down

Classification	Support function for video control service		
Function	Check whether or not to cut down	Symbol	Elib_MS_Check_Cutdown
Functional overview			

- It judges whether or not the specified movie file can be cut down and accordingly returns to AP.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Check_Cutdown (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Return value	Type	-	Description
Ret	Int	O	[Normal end] Possible to cut down: ELIB_MS_OK [Abnormal end] Impossible to cut down: ELIB_MS_NG
Remark			

[Check item for judging whether the file can be cut down]

Table 3-3 Check item for judging whether the file can be cut down

Item	Corresponding type	Constraint
Storage format	MP4・ASF	
Video codec	MPEG4・H.263(baseline)	
Moving image size	SubQCIF・QCIF・QVGA	Except for the prescribed size, it returns ELIB_MS_NG.
Moving image bit rate	768kbps	For the bit rate exceeding 768kbps, it returns ELIB_MS_NG.
Audio codec	AMR・G.726	For AAC, due to hard limit, it returns ELIB_MS_NG.
AMR bit rate	4.75to12.2kbps, with silence compression	
G.726 bit rate	16・24・32kbps	

- Make judgement based on the video information registered in [Video Information Registration].
- QTT codec is incompliant. Even if it exists, it is to be without text, so it is ignored in the processing for the judgement.
- The movie file unavailable for seek will not be cut down.

2.58 Acquisition Of Approximate Value For Cut Down Size

Classification	Support function for video control service		
Function	Get approximate value for cut down size	Symbol	Elib_MS_Get_Cutdown_size

Functional overview

- It returns to AP the approximate value (in KB) for the file size that is created when the specified range is cut down.
- The range is specified from the starting point of time (msec) to the ending point of time (msec).
- From cut down time/video bit rate/video codec/audio bit rate/video frame rate/audio codec, calculate the approximate value for the size of movie data after it is cut down.
- Since the accurate value is figured out only after the encoding is actually completed, the value is to be the approximate value in KB (kilo byte).
- The file will certainly make MP4 file after it is cut down, so the approximate value for cut down size should be in MP4 equivalent.
- It is required that [5. Codec/image quality during editing] [6. Codec/audio quality during editing] of movie editing service attributes be set. If they are not set, the approximate value is calculated by default value of movie editing service attributes.

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_Get_Cutdown_size (unsigned int Ap_ID, int ClipID, unsigned int StartTime, unsigned int EndTime, unsigned int *Size);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
StartTime	unsigned int	I	Cut down start time(msec)
EndTime	unsigned int	I	Cut down end time(msec)
Size	unsigned int*	O	Approximate value for cut down size (in KB)
Return value	**Type**	**-**	**Description**
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

- Nothing in particular.

2.59 Judgement Of Availability For After-Recording

Classification	Support function for video control service		
Function	Check for availability for after-recording	Symbol	Elib_MS_Check_PostRecording
Functional overview			

- It checks if the file with the video information registered is available for after-recording editing and returns the result to AP.
- If the estimated rate for the after-recording result movie size exceeds 800KB, it returns to AP what KB of original movie data for after-recording could fall within the range of 800KB.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Check_PostRecording (unsigned int Ap_ID, int ClipID, int *Size);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Size	int*	O	When the estimated rate for after-recording result movie size exceeds800KB, it returns the limit size of the original movie data for after-recording.
Return value	Type	-	Description
Ret	Int	O	[Normal end] Available for after-recording: ELIB_MS_OK [Abnormal end] Unavailable for after-recording: ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Over 800KB size : ELIB_MS_AF_SIZEOVER

Remark

[Check item for judging whether after-recording can be done]

Table 3-4 Check item for judging whether after-recording can be done

Item	Corresponding type	Constraint
Storage format	MP4	
Video codec	MPEG4・H.263(baseline)	For H.263, anything other than Vendorinformation(3gpp) at the time of shooting by your terminal cannot be edited.
Moving image size	~QCIF	For the size exceeding QCIF, it returns ELIB_MS_NG.
Moving image bit rate	~64.kbps	For the bit rate exceeding 64.kbps, it returns ELIB_MS_NG. Bit rate value for actual data remains the same as the original one for editing, but video information bit rate value is

		recorded by rounding up as follows. ~ 16kbps: 16kbps 17to24kbps: 24kbps 25to32kbps: 32kbps 33to40kbps: 40kbps 17to48kbps: 48kbps 49to56kbps: 56kbps 57to64kbps: 64kbps
Audio codec	AMR	For AAC, due to hard limit, it returns ELIB_MS_NG. Also for AMR, anything other than Vendorinformation(3gpp) at the time of shooting by your terminal cannot be edited.
AMR bit rate	6.7kbps,12.2kbps	

- Make judgement based on the video information registered in [Video Information Registration].
- QTT codec is incompliant. Even if it exists, it is to be without text, so it is ignored in the processing for the judgement.

2.60 Judgement Of Availability For QTT Editing

Classification	Support function for video control service		
Function	Check for availability with QTT editing	Symbol	Elib_MS_Check_QTTEdit
Functional overview			

- It checks if the file with the video information registered is available for QTT editing and returns the result to AP.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Check_QTTEdit (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Return value	Type	-	Description
Ret	Int	O	[Normal end] Available for QTT editing: ELIB_MS_OK [Abnormal end] Unavailable for QTT editing: ELIB_MS_NG
Remark			

[Check item for judging whether QTT editing can be done]

Table 3-5 Check item for judging whether QTT editing can be done

Item	Corresponding type	Constraint
Storage format	MP4	
Video codec	MPEG4・H.263(baseline)	For H.263, anything other than Vendorinformation(3gpp) at the time of shooting by your terminal cannot be edited.
Moving image size	~QCIF	For the size exceeding QCIF, it returns ELIB_MS_NG.
Moving image bit rate	~64kbps	For the rate exceeding 64kbps, it returns ELIB_MS_NG. Bit rate value for actual data remains the same as the original one for editing, but video information bit rate value is recorded by rounding up as follows. ~ 16kbps: 16kbps 17to24kbps: 24kbps 25to32kbps: 32kbps 33to40kbps: 40kbps 17to48kbps: 48kbps 49to56kbps: 56kbps 57to64kbps: 64kbps
Audio codec	AMR	For AAC, due to hard limit, it returns ELIB_MS_NG. Also for AMR, anything other than

		Vendorinformation(3gpp) at the time of shooting by your terminal cannot be edited.
AMR bit rate	6.7kbps,12.2kbps	
QTT codec	tx3g	

- Make judgement based on the video information registered in [Video Information Registration].
- The ones other than the files recorded by your terminal can also be specified but the files other than the ones recorded by your terminal are data-dependent, therefore the editing result will not be guaranteed.

2.61 Acquisition Of Information About Whether Movie Ringtone Is Settable

Classification	Support function for video control service		
Function	Get information about whether movie ringtone is settable	Symbol	Elib_MS_GetCallingLink

Functional overview

- Get the information about whether or not the movie data which information is already registered is available for setting movie ringtone.

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_GetCallingLink(unsigned int Ap_ID, int ClipID, int *calling_link);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
calling_link	int*	I/O	Availability for setting ringtone ELIB_MS_LINK_OK: Available for setting ringtone ELIB_MS_LINK_NG: Unavailable for setting ringtone

Return value	Type	I/O	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- If [Discontinuous memory] is specified when registering video information, this function will generate an error.

2.62 Play Start

Classification	Support Functions of the Video Control Service		
Function Name	Replay Start	Symbol	Elib_MS_Play_Movie
Functional Overview			

- Start playing the video data of the designated Video Information Number.
- If the start position is designated, start Replay from the designated Start Time.
- By setting the Image output destination, the Image output destination of the video that conducts Replay can be designated.
- By setting the Audio output destination, the Audio output destination of the video that conducts Replay can be designated.

The actual storage pointer of the Durface ID is secured at AP.
During this Function Call Complete, the ID is not written in the area for storing the Surface ID input.The Write timing is stored when the Top Image Replay Preparation Complete Event is received.
- Before executing this interface, be sure to execute Replay Sub-Setting.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_Play_Movie (unsigned int Ap_ID, int ClipID, unsigned long Offset, int Video_ Out, int Audio_Out, unsinged int* Surface_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video Information Number
Offset	unsigned long	I	- Start Replay from the designated Start Position (Time Designation in units of msec.)
Video_Out	Int	I	Image Output No Output : ELIB_MS_VOUTP_NONE Own Terminal : ELIB_MS_VOUTP_SELF Recipient Terminal : ELIB_MS_VOUTP_VS
Audio_Out	Int	I	Audio Output No Output : ELIB_MS_AOUTP_NONE Own Terminal : ELIB_MS_AOUTP_SP Recipient Terminal : ELIB_MS_AOUTP_VS
Surface_ID	unsigned int *	O	A pointer to return the Surface ID
Return value	Type	I/O	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG Abnormal Sub-Parameter : ELIB_MS_SUBPARAM_NG Data Not Registered : ELIB_MS_DATANOT

Remark

- Notify that the Replay Preparation is complete from Replay Start using the Top Image Replay Complete Notification Event.The application conducts "D-9 Overlay Setting" and "D-11 Synchronous Display" in that timing and by conducting "D-3 Pause SetRelease", the actual Replay starts.

Only the AP that started the Replay can use the Replay Related Functions from Replay Start to Replay Complete, therefore if the Replay Start function is called redundantly, return an error.Also, if Replay Start functions from other applications are called during Replay, return error.

- If "Non-Continuous Memory" is designated during Video Information Registration, and if the Offset setting is anything other than "0", it is an error.
- If the Replay Flag is set at ELIB_MS_ON in the Replay Sub-Setting, Replay is started from the top after the video data is Replay Complete.
  The Replay does not stop automatically during Loop Replay.
- The Replay Time Notification (MSNotify_PLAY_TIME) during Loop Replay is reset at 0 when it hase Replay to the end and returned to the top.

-If the Replay Sub-Setting is Abnormal Complete or not executed before this interface is executed, return ELIB_MS_SUBPARAM_NG (Abnormal Sub-Parameter).

2.63 Play End

Classification	Support function for video control service		
Function	Play end	Symbol	Elib_MS_Stop_Movie
Functional overview			

- Stop the playing of the movie data of the specified video information number.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Stop_Movie (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Return value	Type	I/O	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

- By calling this function, it notifies the completion of Play stop by Stop completed event (MSNotify_PLAY_STOP).
- If you call this function while the movie is not playing (while you have not called Play start), it returns error.
- If Play stop is called from AP that has not started to play, it returns error.
- If application by which the movie is playing becomes inactive, make sure to call this function and deactivate it.
- If [Discontinuous memory] is specified when registering video information, once the processing for Play is executed, resumption of playing (View thumbnails) is not executable. Delete the video information once and execute it from the video information registration.
- Executing this function will make DSP on stand by in the lower layer. Accordingly the still image recording as well as JPEG stream data working storage used for storage of movie data shot in still image auto continuous recording is released and may become unavailable. In that case, since JPEG stream data working storage cannot be guaranteed by video control, the calling side should withdraw the movie data from JPEG stream data working storage immediately after receiving Still image recording completed event (MSNotify_SNAP_DONE).

2.64 Pause Setting/Release

Classification	Support function for video control service		
Function	Pause Setting/Release	Symbol	Elib_MS_Pause_Movie
Functional overview			

- Set or release the pause during play.
- If you set the pause setting operation while the movie is playing and call this function, it executes the pause processing.
- If you set the pause setting operation while the movie is on pause and call this function, it returns error.
- If you set the pause release operation while the movie is on pause and call this function, it executes the pause release processing.
- If you set the pause release operation while the movie is playing and call this function, it returns error.
- If the movie ELIB cannot accept the pause request while automatically finishing playing, it returns File play finishing: ELIB_MS_PLAYFINISH as return value.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Pause_Movie (unsigned int Ap_ID, int ClipID, int Mode);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Mode	Int	I	Pause setting operation: ELIB_MS_PAUSE_PAUSE Pause release operation: ELIB_MS_PAUSE_RESUME
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG [Pause disabled] File play finishing: ELIB_MS_PLAYFINISH

Remark

- Notify the completion of pause with the movie ELIB by Pause processing completed event (MSNotify_PLAY_PAUSE).
- Notify the completion of resumption of playing with the movie ELIB by Pause processing completed event (MSNotify_PLAY_PAUSE).
- If you call this function while the movie is not playing (while you have not called Play start), it returns error.
- If pause setting/release is called from AP that has not started to play, it returns error.
- Even if [Discontinuous memory] is specified when registering video information, this function can be used only for releasing the pause shortly after the start of playing. In other circumstances, it will not generate errors, but will not guarantee the operation.
- Since the playing speed is maintained during pause, after the pause is released, it plays at the same speed as before the pause setting. (If the playing speed is changed during the pause, it plays at the speed thus changed.)

2.65 Seek

Classification	Support function for video control service		
Function	Seek	Symbol	Elib_MS_Seek_Movie
Functional overview			

- Set the playing position to the specified time.
- If you seek while the movie is playing, it continues to play after completing seek.
- If you seek while the movie is pausing, it pauses after completing seek.
- Seek time is the absolute time specified from the start of movie playing.
- If seek is executed toward the position after the Play end position, it notifies Play End event: MSNotify_PLAY_END instead of returning Seek completed event: MSNotify_PLAY_SEEK, and play will end. (Even if the status before seek request is pause, when seek is executed toward the position after the Play end position, the status will become Play end.)
- If the movie ELIB cannot accept seek request while automatically finishing playing, it returns File play finishing: ELIB_MS_PLAYFINISH as return value.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Seek_Movie (unsigned int Ap_ID, int ClipID, unsigned long clip_begin);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
clip_begin	unsigned long	I	Seek time (millisecond)
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG [Unavailable for seek] File play finishing: ELIB_MS_PLAYFINISH

Remark

- Notify the fact that seek is completed for the movie ELIB by Seek completed event (MSNotify_PLAY_SEEK).
- It does not always play from the specified seek time.
- If you call Seek function before receiving Seek completed event, it returns error.
- If you call this function while the movie is not playing (while you have not called Play start), it returns error.
- If Seek is called from AP that has not started to play, it returns error.
- If [Discontinuous memory] is specified when registering video information, this function will generate an error.

2.66 Change Play Speed

Classification	Support function for video control service		
Function	Change play speed	Symbol	Elib_MS_ChangeSpeed
Functional overview			

- You can change the play speed of movie only while it is playing.
- 0.5 time speed, 1.5 times speed, double speed change is available.
- If the speed is changed to times other than 1.0 time, audio is not output.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_ChangeSpeed (unsigned int Ap_ID, int ClipID, int speed);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Speed	int	I	100% (one time): ELIB_MS_SPEED_NML 200% (two times): ELIB_MS_SPEED_FAST 150% (1.5 times): ELIB_MS_SPEED_SFAST 50% (0.5 time) : ELIB_MS_SPEED_SLOW
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

- If you call this function while the movie is not playing (while you have not called Play start), it returns error.
- If Change play speed is called from AP that has not started to play, it returns error.
- If [Discontinuous memory] is specified when registering video information, this function will generate an error.

2.67 Execute Step

Classification	Support Functions of the Video Control Service		
Function Name	Step Execution	Symbol	Elib_MS_Step_Movie
Functional Overview			

- Advance the paused video a given amount of time (fixed value).
- Audio is not output during Step Execution.
- If Step Execution is conducted after the Replay Complete Position, the Step Complete Event: MSNotify_PLAY_STEP is not returned and the Replay Complete Event: MSNotify_PLAY_END is notified and Replay is complete.(It does not return to the Pause state.)

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_Step_Movie (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video Information Number
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

- Notify that the Step is complete in the Video ELIB using the Step/Frame Advance Completion Event (MSNotify_PLAY_STEP).
- If this function is called when it is not paused, return an error.
- If the Replay Speed Modification is called from an AP that has not conducted Replay Start, return an error.
- If Step is executed before Step/Frame Advance Completion Event is received, return an error.
- If "Non-Continuous Memory" is designated during Video Information Registration, this function is an error.

2.68 View Thumbnails

Classification	Support function for video control service		
Function	View thumbnails	Symbol	Elib_MS_ThumbNail
Functional overview			

- View the thumbnail (the front picture of movie) of the specified video information number.
- Notify the completion of view preparation from view thumbnail request by Play front picture completed (MSNotify_PLAY_FPICT) event. Application performs by that timing [Overlay Setting][D-11 Synchronous Display], and thumbnails are displayed.
- By setting the output to which video picture is output, the output of the video picture for the movie that plays can be specified.
- Into the area (pointer) to store the surface ID input here, the ID is not written when finishing this function call. As for the timing to be written, it is stored at the stage of receiving Preparation for playing front picture completed event.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_ThumbNail (unsigned int Ap_ID, int ClipID, int Video_Out, unsigned int * Surface_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number for viewing thumbnails
Video_Out	Int	I	The output where to output video picture Your terminal : ELIB_MS_VOUTP_SELF Destination terminal : ELIB_MS_VOUTP_VS
Surface_ID	unsigned int*	O	The pointer to return Surface ID
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- Thumbnail view is like the start of movie's playing, and viewing thumbnail while the movie is playing (movie's playing while viewing thumbnail, viewing thumbnail while viewing thumbnail) is not performable.
- If [Discontinuous memory] is specified for clip, this function will not operate. (The request is ELIB_MS_NG.)

2.69 Release of QTT Data

Classification	Support function for video control service		
Function	Release QTT data	Symbol	Elib_MS_ReleaseData
Functional overview			

- Movie control ELIB maintains the data of QuickTimeText emerged in the video data, and sends the pointer to AP as QucikTimeText notification(MSNotify_QTT_DATA). It is the interface to request the release to the movie ELIB in the case that AP does not require the QuickTimeText data any more.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_ReleaseData (unsigned int Ap_ID, int ClipID, void *data);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Data	void *	I	The pointer to the data to be released
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- If the notification of QuickTimeText data is not requested by Elib_MS_Request, video control ELIB must not maintain QuickTimeText data because it cannot accept the release request to QuickTimeText data emerged.
- AP should promptly call QTT data release after having used QuickTimeText data if it is not necessary any more. (It should not be used for many occasions.)
- Before calling Play end processing, make sure to call QTT data release.
- Before termination processing (Elib_MS_Terminate), make sure to call QTT data release.

2.70 Overlay Setting

Classification	Support Functions of the Video Control Service		
Function Name	Overlay Setting	Symbol	Elib_MS_SetOverlay
Functional Overview			

Conduct Overlay setting of the draw area.
AP conducts the settings for everything other than the Video Replay Surface.
- Partial Area Designation of the Input Image
- Draw Area Designation
- Temporary Surface Nondisplay
- Color Key Designation
- Rotation Designation during Draw (Right Direction 90 degrees, Right Direction 180 degrees, Right direction 270 degrees, Flip Horizontal, Flip Vertical)

Include file	Srv_ms_p.h		
Calling Sequence	Int Elib_MS_SetOverlay (unsigned int Ap_ID, Int ClipID, HmiVmOverlayParam *param);		
Argument	Type	I/O	Description
Ap_ID	Unsigned int	I	Application ID
ClipID	Int	I	Video Information Number
Param	HmiVmOverlayParam *	I	Surface Overlay Information Storage Area Pointer
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

AP conducts the Partial Area Setting of the Input Image and the Area Setting of the Draw Area of the Video Surface using the Surface_ID acquired in the Display Resource Acquisition.After conducting setting of the Image Surface, if translucent color drawing is conducted on the Image Surface after calling this API and conducting Overlay Setting, AP displays the image as it is configured.Subsequently, the decoder surface is automatically flipped each time the video decoding is complete, therefore this API is not executed.

2.71 Data Reception Notification

Classification	Support function for video control service		
Function	Data reception notice	Symbol	Elib_MS_ReceiveData
Functional overview			

- In pseudo-streaming and streaming, for the movie ELIB, it notifies to the movie ELIB service that Application has received buffer data.
- Set received data length.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_ReceiveData (unsigned int Ap_ID, int ClipID, char *Rcv_data, unsigned long Rcv_len);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Clip_ID	Int	I	Video information number
Rcv_data	char *	I	Head address of received data (Unused in pseudo-streaming)
Rcv_len	unsigned long	I	Data length of the data additionally received
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Buffer full : ELIB_MS_BUFF_FULL (Discontinuous memory: Occurs only in streaming play)

Remark

- When specifying the continuous memory, if data is added exceeding the buffer size freed up at the start of playing, it returns error.
- If movie data calls the data reception notification at the time other than when continuous memory is specified or when discontinuous memory is specified, it returns error.
- If it is called during the time other than the time between play start and play end, it returns error.
- When specifying the continuous memory, the value of [Rcv_data] is ignored. When specifying the discontinuous memory, set the head address of the data received this time to [Rcv_data].
- In streaming play with discontinuous memory specified, if the receiving rate exceeds the playing rate, the message of received data is stored up in the movie Elib. When exceeding the requested stock volume in the movie Elib, ELIB_MS_BUFF_FULL is set as the return value. (The request has not been accepted.) If Application detects ELIB_MS_BUFF_FULL, it should retry after receiving the acquired data release notification event from the movie Elib.

2.72 Synchronous Display

Classification	Support function for video control service		
Function	Synchronous display	Symbol	Elib_MS_FlipSurf
Functional overview			

- Execute synchronous display.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_FlipSurf (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Nothing in particular

2.73 Acquisition Of Starting Position

Classification	Support function for video control service		
Function	Get starting position	Symbol	Elib_MS_Get_Start_Pos

Functional overview

- Return the current playing position to AP.
- To be executed during the pause is the precondition.
- If it is called at the time other than during the pause, it returns error. (ELIB_MS_NG)
- Cut down start position storage work is freed up byAP.
- If Start_Pos is NULL, it returns ELIB_MS_PARAM_NG.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_Start_Pos(unsigned int Ap_ID, int ClipID, unsigned int *Start_Pos);		
Argument	Type	I/O	Description
Ap_ID	Unsigned int	I	Application ID
ClipID	Int	I	Video information number
Start_Pos	Unsigned int*	O	Cut down start position (msec)
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

- After executing this function, the approximate file size value from the start point will be set to sbinfo of MSNotify_PLAY_TIME(play time notification).

2.74 Acquisition Of End Position

Classification	Support function for video control service		
Function	Get end position	Symbol	Elib_MS_Get_End_Pos

Functional overview

- Return the current playing position to AP.
- To be executed during the pause is the precondition.
- If it is called at the time other than during the pause, it returns error to AP.(ELIB_MS_NG)
- Cut down end position storage work is freed up by AP.
- If End_Pos is NULL, it returns ELIB_MS_PARAM_NG.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_End_Pos (unsigned int Ap_ID, int ClipID, unsigned int *End_Pos);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
End_Pos	unsigned int*	O	Cut down end position (msec)
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- Nothing in particular

2.75 Acquisition Of QTT Sample (Acquisition Of The First 5 Samples)

Classification	Support function for video control service		
Function	Get QTT sample	Symbol	Elib_MS_Get_QTTSample
Functional overview			

- Execute it after receiving The first image play completed event.
- Parse QTT samples for 5 pieces from the top of video information currently playing and store the result in the specified storage space for the acquisition of QTT sample.
- The storage space for the acquisition of QTT sample should be freed up by AP. Since the number of edited characters and effect number are fixed, the size is defined by constant.
- It skips reading QTT data without characters for adjusting the starting time of playing and does not provide to AP.
- If the number of characters of the string exceeds 32 characters, provide the one modified to 32 characters to AP.
- The number defined by the argument QTTNum (maximum 5) is the maximum value.
- The number of QTT samples is less than QTTNum, store from the top of the analysis result structure, and for the rest, set the initial value.
  It returns to AP the number of QTT samples in which character infromation exists as return value.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Get_QTTSample (unsigned int Ap_ID, int ClipID, int QTTNum, _ELIB_MS_QTT_DATA_EDIT * QTTData);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
QTTNum	Int	I	Number of QTT sample structure for edit (1 to 5)
QTTData	_ELIB_MS_QTT_DATA_EDIT*	I/O	QTT sample structure array head address for edit
Return value	Type	-	Description
ret	Int	O	[Normal end] The number of acquired samples QTT: 0 to 5 [Abnormal end] Abnormal end : ELIB_MS_NG Short of size : ELIB_MS_NG_SIZESHORT Parameter abnormal : ELIB_MS_PARAM_NG
Remark			

- Definition of QTT sample structure size for edit

```
ELIB_MS_QTT_DATA_EFSIZE_TEXT  /* Maximum size of QTT string (32 characters (68Byte)) */
ELIB_MS_QTT_DATA_EFSIZE_STYLE/* Maximum size of styling effect (1StyleAtom (32Entry))*/
ELIB_MS_QTT_DATA_EFSIZE_BLINK /* Maximum size of blink effect (1BlinkAtom)            */
```

- QTT sample structure for edit

```
typedef struct tag_ELIB_MS_QTT_DATA_EDIT{
    unsigned long                 PlayTime;      /* Play start time                                  /*
    _ELIB_MS_MP4_TEXT_INFO*MP4TextInfo;  /* QTT header information pointer   /*
    _ELIB_MS_QTT_DATA         *MP4TextData;  /* QTT data analysis result pointer  /*
    int                           Tx;            /* Text track size (Upper left coordinate X: Pixel unit)  /*
    int                           Ty;            /* Text track size (Upper left coordinate Y: Pixel unit)  /*
    int                           TrackWidth;    /* Text track size (Horizontal width: Pixel unit)         /*
    int                           TrackHeight;   /* Text track size (Height: Pixel unit)                   /*
}_ELIB_MS_QTT_DATA_EDIT;
```

- Since the text track size is unique throughout the entire QTT, the same value is set to all the text track sizes of QTT samples acquired.
- For _ELIB_MS_MP4_TEXT_INFO structure, see [Acquisition Of QTT Header Information (MP4)].
- For _ELIB_MS_QTT_DATA structure, see [QuickTimeText Structure].
- Since play start time is not included in both QTT header information and QTT data analysis result, add play start time (Play Time) member.
- AP performs editing for the [Play start time/QTT header information/QTT data analysis result], and will display the text on LCD based on that editing.
- For playing during QTT editing work, directly execute QTT data release (Elib_MS_ReleaseData) without using QTT data (MSNotify_PLAY_QTTDATA) that AP received from the movie ELIB.
- By setting QTT sample acquisition structure obtained here to [Start QTT Combining], the insertion of QTT will be made.

In [Start QTT Combining], QTT sample of NULL is added for adjusting Play Start/End Time. In that case, samples of up to 6 are added and are created as the file of 11 samples in total. When you get 5 samples from this file, get 5 samples excepting the QTT sample of NULL.

2.76 Acquisition Of Play Status

Classification	Support function for video control service		
Function	Get play status	Symbol	Elib_MS_GetPlayStatus
Functional overview			

- Get the play status of the specified video information number.
- If Play end status (MSNotify_PLAY_END has been notified), ELIB_MS_PSTAT_END is set to PlayStatus.
- If the status is other than Play end status (MSNotify_PLAY_END has been notified), a value other than ELIB_MS_PSTAT_END is set to PlayStatus.
  .

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_GetPlayStatus (unsigned int Ap_ID, int ClipID, int * PlayStatus);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
PlayStatus	int*	O	Video control ELIB in status on play ELIB_MS_PSTAT_END: Play end status Other value : Other status
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG

Remark

- If you call this function while the movie is not playing (while you have not called Play start), it returns error.
- If it is called from AP that has not started to play, it returns error.

2.77 Stop Movie (No Flip)

Classification	Support function for video control service		
Function	Stop movie (no flip)	Symbol	Elib_MS_Stop_Movie_NoFlip
Functional overview			

- Stop the playing status of the movie data of the specified video information number.
- Not fill in the screen with the background color when stopping the movie.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Stop_Movie_NoFlip(unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number
Return value	Type	I/O	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

- By calling this function, it notifies the completion of Play stop by Stop completed event (MSNotify_PLAY_STOP).
- If you call this function while the movie is not playing (while you have not called Play start), it returns error.
- If Play stop is called from Application that has not started to play, it returns error.
- If application by which the movie is playing becomes inactive, make sure to call this function and deactivate it.
- If [Discontinuous memory] is specified when registering video information, once the processing for Play is executed, resumption of playing (View thumbnails) is not executable. Delete the video information once and execute it from the video information registration.
- If you want to fill in the screen with the background color when stopping the movie, execute [Play End] instead of using this API.

2.78 Start Recording (Movie Recording)

Classification	Support Functions of the Video Control Service		
Function Name	Record Start	Symbol	Elib_MS_Start_Record
Functional Overview			

- Start recording the video.For Record File Format only MP4 can be designated.
The VIdeo File Record of the designated memory or folder number, and the file number or File Path Direct Designation is possible.
As the Image Input, images from the camera, the TV phone caller image, Avatar Input, and no input can be designated.
- As the Audio Input, the mic input, the caller audio (during TV phone), the caller audio (during TV phone) + mic input, and no input can be designated.
- If there is no input for both the image and audio input, return an error.(ELIB_MS_PARAM_NG)
- It is possible to designate the Replay Restriction Information of the Video file that makes these records.
- If a Record Request occurs during recording, return an error to the AP that called afterwards.(ELIB_MS_NG)
- If the Image Input is a TVPhone Caller Image, and the Image Resolution designated in the service attribute is not QCIF, return an error to AP. (ELIB_MS_NG)
- If the SD card is designated as the Information Storage Location, conduct error check of the SD card and if there is an error, return the error status.

Include file	srv_ms_p.h
Calling Sequence	int Elib_MS_Start_Record (unsigned int Ap_ID, _ELIB_MS_LOCATION *Location, int Video_In, int Audio_In, _ELIB_MS_LIMITINFO *LimitInfo _ELIB_MS_SUBINFO *SubInfo);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Location	_ELIB_MS_LOCATION *	I	Information Storage Location Video Data (Location) * Refer to the Remarks of Video Information Registration for the Structure and the Designation Methods.
Video_In	int	I	Image Input ELIB_MS_CAMERA1 : Camera Input Image (Front) ELIB_MS_CAMERA2 : Camera Input Image (Back) ELIB_MS_MULTENC : TV Phone Caller Image ELIB_MS_VIDEO_AVA : Avatar Input Image ELIB_MS_VIDEO_NONE : No Input (Do not record)
LimitInfo	_ELIB_MS_LIMITINFO *	I	Video Restriction Information (Refer to "B-1 Video Data Registration" for the setting values)
Return value	**Type**	**-**	**Description**
Ret	int	O	[Normal Completion] Normal Completion : ELIB_MS_OK

			[Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG [SD Card Error Status] SD Card Function IO Error : ELIB_MS_SDFIL_EIO SD Card Not Inserted : ELIB_MS_SDFIL_ENOCARD An SD card that exceeds 256MB has been inserted. : ELIB_MS_SDFIL_EUNKNOWNSIZE Write Protect : ELIB_MS_SDFIL_EWRTPRT Format Defect : ELIB_MS_SDFIL_EFORMAT FAT Area Damage : ELIB_MS_SDFIL_EBADFAT Memory Full : ELIB_MS_SDFIL_EDSKFUL System Area Damage : ELIB_MS_SDFIL_EBADSYS Other errors : ELIB_MS_SDFIL_ERROR
Remark			

[Designation Method of the Registration Destination Data
The Designation Method of the Registration Data refers to the Designation Method of the Registration Source Data in the "C-1 Video Information Registration".
* When recording, the "Non-Continuous Memory" cannot be designated.If it is designated, an Abnormal Parameter is returned. （ELIB_MS_PARAM_NG）

[Concerning the Service Attribute Settings during Video Record]
The various parameters set in "Service Attribute Setting (or All Service Attribute Settings)" conducted during Video Record and before Video Record Start have the following restrictions.

- The File Format during recording and restrictions concerning image/audio quality
The values that can be set for each item of every File Format

Table 4.2 Values that can be set for each item of every File Format

			MP4／MP4ST
Image Quality	codec_type	Codec Type	ELIB_MS_MEDIA_V_MPG4 ELIB_MS_MEDIA_V_H263 ELIB_MS_MEDIA_V_H263BL
	resolution	Image Resolution	ELIB_MS_RSL_SUBQCIF ELIB_MS_RSL_QCIF
	rate	Target Bit Rate Bit Rate Designation	Target Bit Rate ELIB_MS_VREC_MP4_FINE ELIB_MS_VREC_MP4_NOMAL ELIB_MS_VREC_MP4_ECN
	br_324	324 Encode Rate	Setting Not Possible
Audio Quality	codec_type	Codec Type	ELIB_MS_MEDIA_A_AMR
	format_type	Record Format	ELIB_MS_AFORM_IF2
	Bitrete	Target Bit Rate	ELIB_MS_ABITRZ_LV3 (6.7kbps) ELIB_MS_ABITRZ_LV7 (12.2kbps)

	VOX_flag	Compression	ELIB_MS_NCMP_NUSE (No Sound Compression Not Used)

* Operations for when the setting value designated is not one of the above: The return value of the function is ELIB_MS_OK but the record is a failure.

[Concerning the Setting of Replay Restriction Information/Registration Contents Information] If the ELIB_MS_LOC_MEM (Continuous Memory) is designated in the Location Type of the ClipLocation (Information Storage Source), Replay Restriction Information is not secured, therefore be sure to set the Replay Restriction Type of the Replay Restriction Information to No Replay Restriction.
Concerning SubInfo (Registration Contents Sub-Information) information, acquire the Setting Information (UIM Identifier ID and Size) at the Request side AP beforehand.(In the Video Control ELIB, the content of the UIM Identifier ID is not recognized.)

2.79 Stop Recording

Classification	Support function for video control service		
Function	Stop recording	Symbol	Elib_MS_Stop_Record
Functional overview			

- Stop recording the movie and store the data for up to the stop.
- If the recording is stopped from other than AP that has started to record, it returns error instead of stopping the recording. (ELIB_MS_NG)

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Stop_Record (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Normal end] Abnormal end : ELIB_MS_NG

Remark

- If Stop Recording is requested, it notifies the fact that the recording processing has ended by Recording end (End condition established) event (MSNotify_REC_END) to the Appli. If the Appli receives the above event, it should execute the start of post-processing of movie recording ([See Start Post-Processing Of Movie Recording]) after having executed the processing for waiting the post-processing in the Appli such as view as pop up. Movie control ELIB notifies recording End (post-processing completed) event (MSNotify_REC_DONE)) to the Appli when it completes the post-processing.
- If this function is called after [Start Post-Processing Of Movie Recording] has been executed, it returns ELIB_MS_NG and does not accept the processing.

2.80 Cancel Recording

Classification	Support function for video control service		
Function	Cancel recording	Symbol	Elib_MS_Cancel_Record

Functional overview

- Cancel the record of movie and discard the data in progress.
- If the recording is cancelled from other than AP that has started to record, it returns error instead of cancelling the recording. (ELIB_MS_NG)

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Cancel_Record (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- By calling this function, it notifies the fact that Cancel recording is completed by Cancel recording completed event (MSNotify_REC_CANCEL).
- If Recording end event had occurred before the execution of recording cancel by movie ELIB, it returns ELIB_MS_NG to the calling AP and does not execute the processing for the cancellation. Also, if this function is called after [Start Post-Processing Of Movie Recording] has been executed, it returns ELIB_MS_NG to the calling AP and does not accept the processing.

2.81 Still Image Recording

Classification	Support Functions of the Video Control Service		
Function Name	Still Image Record	Symbol	Elib_MS_Start_Snap
Functional Overview			

- Record the Still Image and get the data in the memory.
- The Still Image Record only records on the memory.
- The output is "Uncompressed (YUV422)" and "Uncompressed (RGB565)" in accordance to the Service Attribute Values.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_Start_Snap (unsigned int Ap_ID, int From, BYTE *Buffer, int Size);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
From	int	I	Designate input of Camera, Decoded Images and the like ELIB_MS_CAMERA1 : Camera Input Image (Front) ELIB_MS_CAMERA2 : Camera Input Image (Back) ELIB_MS_DECODER : Record from Replay Image
Buffer	BYTE*	I	Area to save the recorded data. (Physical Address of the JPEG Stream Data Operation Area)
Size	int	I	Size of Storage Area (unit: bytes)
Return value	Type	-	Description
Ret	int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

- The buffer for Still Image Record must be secured on the Caller side.
- The recorded results are returned to asynchronous as the Still Image Record Complete Event (MSNotify_SNAP_DONE).The size of the data recorded in the event is stored.
- The format of the recorded Still Image Data is not converted by the Video Elib, therefore the supported format depends on the driver that is mounted in the subordinate.
  -> The Camera Image "YUV422", TV Phone Image, and Replay Image "Uncompressed (RGB565)" can be acquired.

Due to the specifications of this function, Still Image Acquisition in the designated format is possible, however, this specification may be limited because of the model mounting.
Mobile phones have the fixed output above and the Still Image Format Designation is in substance fixed by the Input Source.

- If the Data Format that is returned from the Sub-driver (Video Handler) for the same acquisition function due to Camera change (JPEG camer =. YCbCr camera) and the like, the Image Format that is output depends on the change (YCbCr data is returned even when JPEG is designated.).However, the operation of this function when changes like that occur is not guaranteed.

- The format of the Still Image Data to be stored changes depending on the value designated in the input of Camera, Decode Image and the like.
- The area where the recorded data is stored (Buffer), designate the physical address of the JPEG Stream Data Operation Area.
  * Concerning the JPEG Stream Data Operation Area, refer to "DSP lib Interface Specification" and conduct Secure and Release.
  Additionally, the JPEG Stream Data Operation Area may be not usable when the DSP is in the Standby condition (such as during the Video Replay Complete Process by Elib_MS_Stop_Movie).In such cases, for Video Control, the JPEG Stream Data Operation Area cannot be secured, therefore the caller side pulls up the Image Data from the JPEG Stream Data Operation Area immediately after receiving the Still Image Record Complete Event (MSNotify_SNAP_DONE).

2.82 Cancel Still Image Recording

Classification	Support function for video control service		
Function	Cancel still image recording	Symbol	Elib_MS_Cancel_Snap

Functional overview

- Stop recording still image.
- If the still image recording is cancelled from other than AP that has started to record the still image, it returns error instead of cancelling the still image recording. (ELIB_MS_NG)

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Cancel_Snap (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- By calling this function, it notifies the fact that Cancel still image recording is completed by Cancel still image recording completed event (MSNotify_SNAP_CANCEL).
- If Still image recording end event had occurred before the execution of Still image recording cancel, it returns ELIB_MS_NG to the calling AP and does not execute the processing for the cancellation.

2.83 Acquisition Of Still Image Information

Classification	Support Functions of the Video Control Service		
Function Name	Still Image Information Acquisition	Symbol	Elib_MS_GetPictureInfo
Functional Overview			

- For Still Image Data of multiple Still Image Records, when a Still Image Number is designated, the Still Image Data size and the Storage Destination Address is returned.
- If the Still Image Numbers above that are stored are designated, return an error.

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_GetPictureInfo (unsigned int Ap_ID, int PictNum, BYTE* SnapData, int* PictSize, BYTE** Buffer,);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
PictNum	int	I	Still Image Storage Number
SnapData	BYTE*	I	Top Pointer of the Still Image Data (Multiple Still Images)
PictSize	int *	O	Still Image size of the designated number
Buffer	BYTE**	O	Top Pointer of the Designated Still Image Data
Return value	Type	-	Description
Ret	int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

- The Storage Number of the Still Image should always be set at 1.

2.84 Start Post-Processing Of Movie Recording

Classification	Support function for video control service		
Function	Start post-processing of movie recording	Symbol	Elib_MS_Finish_Record

Functional overview

- After receiving the event notification [Recording End (End Condition Established)] from the movie control ELIB, it requests the start of post-processing to the movie control ELIB.
- If the recording is stopped from other than AP that has started to record, it returns error instead of stopping the recording. (ELIB_MS_NG)

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Finish_Record (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- After executing the start of post-processing for movie recording, the video control ELIB executes the processing (post-processing) for writing the header etc. into the file. The fact that the post-processing was completed is notified to the Appli by Recording end (post-processing completed) event (MSNotify_REC_DONE).

* This processing is the API that was added in order to deal with the problem that it takes 1 to 2 seconds for the pop-up to be viewed after the end of recording, which is the task with P2102V. Appli should execute this API without fail after the completion of the necessary operations such as view as pop up after having received the event [Recording End (End Condition Established)] from the video control ELIB.

- This API will become enabled only after the reception of [Recording End Notice (End Condition Established)] from the video control Elib after having executed [Start Recording (Movie Recording)].
- After executing this API, Appli will not accept [Stop Recording] [Cancel Recording]. (It continues the post-processing operation.)

2.85 Start Cut Down

Classification	Support function for video control service		
Function	Start cut down	Symbol	Elib_MS_Start_Cutdown

Functional overview

- Start to cut down the movie data of the specified video information for the specified range covering from the start position to the end position.
- In the case of cutting down up to the Play end position by getting only the start position without getting the end position, set total play time to the end position.
- The output memory of cut down data should be secured in AP.
- The entity of playing surface ID storage pointer for the data from which to cut down should be secured by AP.
- The movie file from which to edit should be with the video information registered.
- The type of codec for the data that is created by cut down should be specified by setting the video editing service attributes for [codec/image quality] [codec/audio quality] before starting to cut down.
- Into the area to store the surface ID input here, the ID is not written when finishing this function call. As for the timing to be written, it is stored at the stage of receiving Edit front picture play completed notification (MSNotify_EDIT_FPICT).
- Processing for cutting down is executed in [Real time conversion] by which playing and recording are simultaneously performed.
- This function is performable only with the files in SD card, Flash and on the continuous memory. In other cases than that, it returns error (ELIB_MS_PARAM_NG).

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_Start_Cutdown (unsigned int Ap_ID, int ClipID, BYTE *Data, int Size, unsigned int *Data, unsigned int End_pos, unsigned int * Surface_ID);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number for the file from which to cut down
Data	BYTE *	O	Cut down data storage area head address
Size	int	I	Size of the area (Data) freed up by AP
Start_pos	unsigned int	I	Start cut down from the specified start position (the time specified in millisecond).
End_pos	unsigned int	I	End cut down at the specified end position (the time specified in millisecond).
Surface_ID	unsigned int*	O	The pointer to return Surface ID
Return value	**Type**	**-**	**Description**
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- Enlargement of image size after cutting down (SubQCIF->QCIF/SubQCIF->QVGA/QCIF->QVGA) is error.

2.86 Stop Cut Down

Classification	Support function for video control service		
Function	Stop cut down	Symbol	Elib_MS_Stop_Cutdown
Functional overview			

- Stop cutting down the movie and store the data for up to the stop.
- If cut down is stopped from other than AP that requested the start of cut down, it returns error instead of stopping cut down. (ELIB_MS_NG)

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Stop_Cutdown (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to cut down
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

- If Stop cut down is requested, it notifies the fact that the cut down processing has ended by Edit end (End Condition Established) event (MSNotify_EDIT_END) to Appli.
- Then, it notifies the fact that the processing (post-processing) for writing information such as header into the file was completed after having stopped by Edit end (post-processing completed) event (MSNotify_EDIT_DONE) to Appli.

2.87 Cancel Cut Down

Classification	Support function for video control service		
Function	Cancel cut down	Symbol	Elib_MS_Cancel_Cutdown

Functional overview

- Cancel cut down and discard the data in progress of cutting down.
- If the cancel of cut down is requested from other than AP that requested the start of cut down, it returns error instead of executing the cancel of cut down. (ELIB_MS_NG)

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Cancel_Cutdown (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to cut down
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

- It notifies the fact that Cancel cut down was completed by Cancel edit completed event (MSNotify_EDIT_CANCEL).
- If writing into the file had ended when this function was called, it returns ELIB_MS_NG to the calling AP and does not execute the cancel processing.

2.88 Start After-Recording

Classification	Support function for video control service		
Function	Start after-recording	Symbol	Elib_MS_Start_PostRecording
Functional overview			

- The output memory of the after-recording edited data should be secured in AP.
- The entity of playing surface ID storage pointer for the original data for after-recording edit should be secured by AP.
- The type of codec which is created by after-recording will be [codec/image quality] [codec/audio quality] that is the same as the file for after-recording.
- Into the area to store the surface ID input here, the ID is not written when finishing this function call. As for the timing to be written, it is stored at the stage of receiving Edit front picture play completed notification (MSNotify_EDIT_FPICT).
- The movie file from which to edit should be with the video information registered.
- Processing for after-recording is executed in [Real time conversion] by which playing and recording are simultaneously performed.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Start_PostRecording (unsigned int Ap_ID, int ClipID, BYTE *Data, int Size, int Audio_In, unsigned int * Surface_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
Data	BYTE *	O	Edit data storage area head address
Size	int	I	Size of the area (Data) freed up by AP
Audio_In	int	I	Initial value for audio input Mike input : ELIB_MS_AINP_MIC Audio during play : ELIB_MS_AINP_PLAYA
Surface_ID	unsigned int*	O	The pointer to return Surface ID
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- Notify the fact that the recording preparation has been completed from Start audio recording of after-recording by Edit front picture play completed notification (MSNotify_EDIT_FPICT). Application executes by that timing [Edit Overlay Setting], [Synchronous Edit Display], and the actual audio recording of after-recording is started by executing [Release Edit Pause].
- When starting after-recording, put a stop to playing by using [Play End].

2.89 Change After-Recording Audio

Classification	Support function for video control service		
Function	Change after-recording audio	Symbol	Elib_MS_Change_PostRecording_Audio
Functional overview			

- Change the record audio input of after-recording audio.
- If mike voice input is set while inputting mike voice, it executes nothing and returns Normal end. (ELIB_MS_OK)
- If movie file audio input is set while inputting movie file audio, it executes nothing and returns Normal end. (ELIB_MS_OK)
- Call this function while editing after-recording. (If it is called in other status, it returns error.)

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Change_PostRecording_Audio (unsigned int Ap_ID, int ClipID, int Mode);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
Mode	int	I	Mike voice input : ELIB_MS_AINP_MIC Movie file audio input : ELIB_MS_AINP_PLAYA
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal: ELIB_MS_PARAM_NG

Remark

- Nothing in particular.

2.90 Cancel After-Recording

Classification	Support function for video control service		
Function	Cancel after-recording	Symbol	Elib_MS_Cancel_PostRecording

Functional overview

- Cancel after-recording and discard the after-recording data in process of creation.
- If after-recording is cancelled from other than AP that requested to start after-recording, it returns error instead of cancelling after-recording. (ELIB_MS_NG)
- Call this function while editing after-recording. (If it is called in other status, it returns error.) (ELIB_MS_NG)
- It notifies the fact that Cancel after-recording has been completed by Cancel edit completed event (MSNotify_EDIT_CANCEL).
- If Edit end (post-processing completed) event had occurred before the execution of cancelling after-recording by movie ELIB, it returns ELIB_MS_NG to the calling AP and does not execute the cancel processing.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Cancel_PostRecording (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Nothing in particular.

2.91 Stop After-Recording

Classification	Support function for video control service		
Function	Stop after-recording	Symbol	Elib_MS_Stop_PostRecording
Functional overview			

- Call this function while editing after-recording. (If it is called in other status, it returns error.)
- It notifies the fact that after-recording processing has been completed by Edit end (End Condition Established) event (MSNotify_EDIT_END) to AP.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Stop_PostRecording (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Nothing in particular.

2.92 Start End Processing Of After-Recording

Classification	Support function for video control service		
Function	Start end processing of after-recording	Symbol	Elib_MS_End_PostRecording
Functional overview			

- Execute the processing for recording the audio data for after the position stopped at the time when you had stopped after-recording.
- Call this function while editing after-recording (after notifying Edit End (End Condition Established)). (If it is called in other status, it returns error.)
- After executing this processing, it sends Edit status notification (MSNotify_EDIT_TIME) to AP at intervals.
- After completing end processing of after-recording, it notifies Edit end (post-processing completed) (MSNotify_EDIT_DONE) to AP.
- When sending Edit status notification (MSNotify_EDIT_TIME) to AP, it suspends the end processing. AP executes [F-9 Resume End Processing Of After-Recording] and resumes the end processing.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_End_PostRecording (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Nothing in particular.

2.93 Resume End Processing Of After-Recording

Classification	Support function for video control service		
Function	Resume end processing of after-recording	Symbol	Elib_MS_End_Resume_PostRecording

Functional overview

- Resume End processing of after-recording that was suspended when Edit Status notification was sent to AP by End processing of after-recording.
- After executing this processing, it sends Edit status notification (MSNotify_EDIT_TIME) to AP at intervals.
- After completing end processing of after-recording, it notifies Edit End (post-processing completed) (MSNotify_EDIT_DONE) to AP.
- When sending Edit status notification (MSNotify_EDIT_TIME) to AP, it suspends the end processing. AP again executes [Resume End Processing Of After-Recording] and resumes the end processing.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_End_Resume_PostRecording (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Nothing in particular.

2.94 Start QTT Combining

Classification	Support function for video control service		
Function	Start QTT combining	Symbol	Elib_MS_Start_QTTEdit

Functional overview

- Start the processing for combining the input QTT sample structure array with the corresponding file.
- The output memory after combining the QTT sample should be secured by AP.
- The movie file from which to edit should be with the video information registered.
- The type of codec which is created by QTT combining will be [codec/image quality] [codec/sound quality] that is the same as the file for inserting QTT sample.
- Get the QTT sample for editing from [Acquisition Of QTT Sample (Acquisition Of The First 5 Samples)].
- After completing the processing for combining the data covering the time specified by edit combining process notification interval, it sends MSNotify_EDIT_TIME (Edit status notification) to AP.
- When it sends Edit status notification to AP, QTT combining processing suspends. AP executes [Resume QTT Combining Processing] and resumes the combining processing.

Include file	srv_ms_p.h
Calling sequence	int Elib_MS_Start_QTTEdit(unsigned int Ap_ID, int ClipID, int QTTNum, _ELIB_MS_QTT_DATA_EDIT *QTTData, _ELIB_MS_QTT_COLORRGBA BackColor, int Size, BYTE *Buff);

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
QTTNum	int	I	Number of QTT sample structure for combining (up to5)
QTTData	_ELIB_MS_QTT_DATA_EDIT*	I	QTT sample structure array head address for edit
BackColor	_ELIB_MS_QTT_COLORRGBA	I	Blank QTT background color for time adjustment
Size	int	I	The output size
Buff	BYTE*	O	The output pointer
Return value	**Type**	**-**	**Description**
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG Parameter abnormal : ELIB_MS_PARAM_NG Unregistered data : ELIB_MS_DATANOT

Remark

- Set the background color on which QTT is not displayed due to a certain blank generating in QTT display interval to blank QTT background color for time adjustment.
- For text track size of _ELIB_MS_QTT_DATA_EDIT structure array, only the value set to the text track size of the head structure is valid.

2.95 Cancel QTT Combining

Classification	Support function for video control service		
Function	Cancel QTT combining	Symbol	Elib_MS_Cancel_QTTEdit
Functional overview			

- Cancel QTT combining and discard the movie data in process of creation.
- Call this function while editing QTT. If it is called in other status, it returns error.
- It notifies the fact that Cancel QTT combining has been completed by Cancel edit completed event (MSNotify_EDIT_CANCEL).
- By recording ID of AP that is editing QTT combining, it compares with ID of QTT combining cancel request AP and if they do not match with each other, it executes error processing.
- If Edit end (post-processing completed) event had occurred before the execution of cancelling QTT combining by movie ELIB, it returns ELIB_MS_NG to the calling AP and does not execute the cancel processing.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Cancel_QTTEdit(unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Nothing in particular.

2.96 Resume QTT Combining Processing

Classification	Support function for video control service		
Function	Resume QTT combining processing	Symbol	Elib_MS_Resume_QTTEdit
Functional overview			

- Resume QTT combining processing that is suspended when Edit status notification has been sent to AP by QTT combining processing.
- After completing QTT combining, it notifies MSNotify_EDIT_DONE (Edit End: post-processing completed) to AP.
- When it sends Edit status notification to AP, QTT combining processing suspends. AP again executes [Resume QTT Combining Processing] and resumes the combining processing.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Resume_QTTEdit (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	int	I	Video information number of the file from which to edit
Return value	Type	-	Description
Ret	int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Nothing in particular.

2.97 Edit Overlay Setting

Classification	Support Functions of the Video Control Service		
Function Name	Edit Overlay Setting	Symbol	Elib_MS_Set_Overlay_Edit

Functional Overview

Conduct Overlay setting of the draw area.
AP conducts the settings for everything other than the Video Replay Surface.
- Partial Area Designation of the Input Image
- Draw Area Designation
- Temporary Surface Nondisplay
- Color Key Designation
- Rotation Designation during Draw (Right Direction 90 degrees, Right Direction 180 degrees, Right direction 270 degrees, Flip Horizontal, Flip Vertical)

Include file	srv_ms_p.h		
Calling Sequence	int Elib_MS_Set_Overlay_Edit (unsigned int Ap_ID, int ClipID, HmiVmOverlayParam_p *param);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video Information Number
param	HmiVmOverlayParam_p *	I	Surface Overlay Information Storage Area Pointer
Return value	Type	-	Description
Ret	Int	O	[Normal Completion] Normal Completion : ELIB_MS_OK [Abnormal Completion] [Abnormal Completion] : ELIB_MS_NG Abnormal Parameter : ELIB_MS_PARAM_NG

Remark

AP conducts the Partial Area Setting of the Input Image and the Area Setting of the Draw Area of the Video Surface using the Surface_ID acquired in the Display Resource Acquisition.After conducting setting of the Image Surface, if translucent color drawing is conducted on the Image Surface after calling this API and conducting Overlay Setting, AP displays the image as it is configured.Subsequently, the decoder surface is automatically flipped each time the video decoding is complete, therefore this API is not executed.

2.98 Synchronous Edit Display

Classification	Support function for video control service		
Function	Synchronous edit display	Symbol	Elib_MS_FlipSurf_Edit
Functional overview			

- Execute synchronous display.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_FlipSurf_Edit (unsigned int Ap_ID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG

Remark

- Nothing in particular

2.99 Release Edit Pause

Classification	Support function for video control service		
Function	Release edit pause	Symbol	Elib_MS_Resume_Edit
Functional overview			

- Start to record while releasing the pause of playing after executing Edit overlay setting and Synchronous edit display after having executed either one of Start cut down/Start after-recording.
- If you call this function in a status other than while the automatic edit play is OFF, it returns error.

Include file	srv_ms_p.h		
Calling sequence	int Elib_MS_Resume_Edit (unsigned int Ap_ID, int ClipID);		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ClipID	Int	I	Video information number
Return value	Type	-	Description
Ret	Int	O	[Normal end] Normal end : ELIB_MS_OK [Abnormal end] Abnormal end : ELIB_MS_NG
Remark			

- Notify the fact that either Start cut down or Start after-recording has been completed by the movie ELIB by Start edit event (MSNotify_EDIT_START).
- If you call this function while either Start cut down or Start after-recording has not been executed, it returns error.
- If it is called from AP that has not executed either Start cut down or Start after-recording, it returns error.

3. Videophone Service
3.1 Start videophone

Classification	Videophone ELIB function		
Function	Start videophone	Symbol	Elib_AV_Start_TV
Functional overview			

Use conditions:
Use this function when videophone communication is not in progress. Do not call this function in any other state.
Send message to the videophone ELIB thread
Instruction to start videophone
Videophone ELIB processing:
Tell 3G324M to start a session.
Set a voice path depending on the open state of the audio channel.

Set the overlay for VAPI.
The completion of the session start processing shall be notified using the session start notification of the videophone state monitoring notification.

Include file	srv_av.h
Calling sequence	int Elib_AV_Start_TV (Ap_ID, Rate , Start_Type, Ton_No)

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Rate	Int	I	Communication rate ELIB_AV_RATE_32 /* 32K connection */ ELIB_AV_RATE_64 /* 64K connection */
Start_Type	Int	I	Videophone start type ELIB_AV_START_KIND_CAMERA /* Self image */ ELIB_AV_START_KIND_SUBPICTURE /* Alternate image */ ELIB_AV_START_KIND_TVMEMO /* Videophone message memo */ ELIB_AV_START_KIND_REMOTE_SURV /* Remote surveillance *1 */ ELIB_AV_START_KIND_RCVHOLD /* Response hold */ ELIB_AV_START_KIND_CAMM_NG /* Camera out of order */ ELIB_AV_START_KIND_REMOTE_SUB /* Remote surveillance (alternate image) *1 */ ELIB_AV_START_KIND_VOICEMEMO /* Voice message memo */ ELIB_AV_START_KIND_REMOTE_CAMNG /* Remote surveillance *5 (camera out of order) */ ELIB_AV_START_KIND_SELSUB_FIX /* Call-by-call alternate image (preinstalled) */ ELIB_AV_START_KIND_SELSUB_USER /* Call-by-call alternate image (custom) */
Tone_No	Int	I	Response message *2 ELIB_AV_TONE_MSG /* Fixed message */ ELIB_AV_TONE_PRIVATE /* Private message */ ELIB_AV_TONE_ENG /* English message */ ELIB_AV_TONE_SPEECH1 /* Talk 1 */ ELIB_AV_TONE_SPEECH2 /* Talk 2 */

			ELIB_AV_OUTOUHORYU_MSG1 /* Response hold tone 1 */ ELIB_AV_OUTOUHORYU_MSG2 /* Response hold tone 2 */
Return value	Type	I/O	
Ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG(parameter error, nonobservance of prerequisite, message sending failure)

Remark

*1: Set remote surveillance only on the side where remote surveillance is received.
During remote surveillance, specify "remote surveillance" if the image to be sent to the remote subscriber is a local subscriber image.

*2: The response message number is valid only if the videophone start type is videophone message memo or videophone response hold.

*3: The response message number shall be changed by the videophone ELIB to either of the following tone types (Audio control file replay API).

- Videophone message memo:
  Standard : ELIB_AV_TONE_MSG → MSL_MSG_PLAY1
  Private : ELIB_AV_TONE_PRIVATE → MSL_MSG_PLAY2
  English : ELIB_AV_TONE_ENG → MSL_MSG_PLAY3
  Talk 1 ELIB_AV_TONE_SPEECH1 (MSL_REC_SPALSTA1
  Talk 2 ELIB_AV_TONE_SPEECH2 (MSL_REC_SPALSTA2
- Videophone response hold:
  Response hold tone 1 ELIB_AV_OUTOUHORYU_MSG1 (MSL_RP_FIX_PAUSE_LOOP
  Response hold tone 2 ELIB_AV_OUTOUHORYU_MSG2 (MSL_RP_FIX_PAUSE_LOOP2
  Talk 1 ELIB_AV_TONE_SPEECH1 (MSL_REC_SPALSTA_LOOP1
  Talk 2 ELIB_AV_TONE_SPEECH2 (MSL_REC_SPALSTA_LOOP2

*4: During remote surveillance, specify this parameter if the camera is out of order. Send the "camera out of order" image to the remote subscriber.

3.2 End videophone

Classification	Videophone ELIB function		
Function	Close videophone	Symbol	Elib_AV_End_TV
Functional overview			

Use conditions:
Nothing in particular.
Send message to the videophone ELIB thread:
Instruction to close videophone
Videophone ELIB processing:
Send 3G324M an instruction to end the session to close the videophone communication.
The completion of the session end processing shall be notified using the session end notification of the videophone state monitoring notification.

Include file	srv_av.h		
Calling sequence	int Elib_AV_End_TV (Ap_ID,Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	Int	I	Normal end: ELIB_AV_END_NORMAL Forced end: ELIB_AV_END_FORCE
Return value	Type	I/O	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (message sending failure)
Remark			

3.3 Set overlay

Classification	Videophone ELIB function		
Function	Set overlay	Symbol	Elib_AV_Set_Overlay

Functional overview

Use conditions:
- As the prerequisite for calling this function, the setting of an image to send to the remote office shall be complete (A notification of setting an image to the remote subscriber shall be received) (The camera shall be turned on before a camera image is sent).
- Call this function if the display layout is changed such as when call hold, response hold, or videophone message memo ends.
- This service cannot be called if a surface by other service is set.
- As a prerequisite, all the overlay images other than graphics shall be available.
- The set overlay function shall not be called again during the overlay setting (until a notification of completion of setting the overlay is received).

Send message to the videophone ELIB thread:
Instruction to set an overlay
Videophone ELIB processing:

Set the overlay for VAPI.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_Overlay (Ap_ID, Dev_ID, Overlay_Inf)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Dev_ID	unsigned int	I	Device ID ELIB_AV_DEV_MAINLCD: Main LCD ELIB_AV_DEV_SUBLCD : Rear LCD ELIB_AV_DEV_AVOUT :AVOUT
Overlay_Inf	_ELIB_AV_OVERLAY_PARAM *	I	Overlay setting information (see below)
Return value	Type	I/O	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error, nonobservance of prerequisite, message sending failure) AV-OUT cable out error: ELIB_AV_CABLEERR

Remark

A completion state shall be acknowledged by the videophone state monitoring.

To use a structure in this function, name the common data defined in VAPI using typedef (changing the beginning of the name from "hmi_" to "_ELIB_").

```
typedef HmiVmOverlayParam_p      _ELIB_AV_OVERLAY_PARAM;
typedef HmiVmOverlayParamSurf_p _ELIB_AV_OVERLAY_PARAM_SURF;
```

* The return value "AV-OUT cable out error: ELIB_AV_CABLEERR" is provided to conform to the P-company interface. The return value will not be returned.

* This function, when called, shall perform the control of event flag [wai_flgx] and put the videophone APL in the wait state.
(When the wait state is cleared, clr_flgx shall be used to clear the event flag.)

3.4 Set image to send to remote subscriber

Classification	Videophone ELIB function		
Function	Set image to send to remote subscriber	Symbol	Elib_AV_Set_Send_Data
Functional overview			

Use conditions:
- This function can be used when videophone connection is being made or videophone communication is in progress.
- This function shall be called when the target to be sent to the remote subscriber is changed.
- When a self image is selected, the camera shall be turned on.
- When the camera is out of order, a camera out of order image shall be sent to the remote subscriber.

Send message to the videophone ELIB thread:
Instruction to set an image to send to the remote subscriber
Videophone ELIB processing:
Send 3G324M a send control instruction. To send a static image (alternate image, hold image, or out of order image), first set a surface and then send a send control instruction.

Include file	srv_av.h
Calling sequence	int Elib_AV_Set_Send_Data(Ap_ID, Type)

Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Type	Int	I	Display type ELIB_AV_DISPTYPE_CAMERA /* Self image */ ELIB_AV_DISPTYPE_SUBPICTURE /* Alternate image */ ELIB_AV_DISPTYPE_TALKHOLD /* Call hold image */ ELIB_AV_DISPTYPE_MOVIE /* Movie file */ ELIB_AV_DISPTYPE_CAMM_NG /* Camera out of order image */ ELIB_AV_DISPTYPE_VOICEMEMO /* Videophone voice message */ ELIB_AV_DISPTYPE_PICTURE /* Static image file */
Return value	Type	I/O	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error, nonobservance of prerequisite, message sending failure)

Remark
A completion state shall be notified by the videophone state surveillance.
When the completion notification for this interface is received, the overlay setting shall be made.

3.5 Switch communication modes

Classification	Videophone ELIB function		
Function	Switch communication modes	Symbol	Elib_AV_Picture_Quality_Change
Functional overview			
Use conditions: Videophone communication is in progress. Send message to the videophone ELIB thread Request to switch communication modes Videophone ELIB processing: Request 3G324M to switch communication modes. Since this mode is valid only when communication is in progress, the communication mode setting value shall not be changed.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Picture_Quality_Change (Ap_ID,Type)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Type	Int	I	Image quality mode ELIB_AV_VIDEO_MOVIE : Movement priority ELIB_AV_VIDEO_NORMAL : Standard ELIB_AV_VIDEO_QUALITY: Quality priority
Return value	Type	I/O	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Parameter error, message sending failure)
Remark			

3.6 Switch hands-free modes

Classification	Videophone ELIB function		
Function	Switch hands-free modes	Symbol	Elib_AV_HandsFree
Functional overview			

Use conditions:
Use this function to enable or disable the hands-free feature while videophone communication is in progress.
The hands-free setting shall be OFF by default when communication is started.
Send message to the videophone ELIB thread:
Notification of switching hands-free modes
Videophone ELIB processing:
Switch hands-free modes to switch voice paths.

Include file	srv_av.h		
Calling sequence	int Elib_AV_HandsFree (Ap_ID,Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	Int	I	ELIB_AV_HANDSFREE_ON: Hands-free enabled ELIB_AV_HANDSFREE_OFF: Hands-free disabled
Return value	Type	I/O	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Parameter error, message sending failure)
Remark			

3.7 Request for event generation notification

Classification	Videophone ELIB function		
Function	Request for event generation notification	Symbol	Elib_AV_ Request
Functional overview			
Use conditions: No limitation Send message to the videophone ELIB thread: None Videophone ELIB processing: Notify a designated event to an application that called this function. No event shall be notified by default.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_ Request (Ap_ID, Mask)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mask	unsigned int	I	Masks for notified events *1 Enabled if the bit is ON and disabled if it is OFF ELIB_AV_MASK_TVSTART 0x0001: Acknowledge videophone start ELIB_AV_MASK_TVTLK 0x0002: Notify videophone start ELIB_AV_MASK_TVEND 0x0004: Notify videophone end ELIB_AV_MASK_CHANNEL_PATH 0x0008: Notify send/receive path state ELIB_AV_MASK_TVSTATUS 0x0010: Notify communication state ELIB_AV_MASK_OVERLAY 0x0020: Notify completion of setting overlay ELIB_AV_MASK_SETSURF 0x0040: Notify completion of setting image to send to remote subscriber ELIB_AV_MASK_FSTAT_CHG 0x0100: Notify change of housing state during videophone communication ELIB_AV_MASK_RSIZE_ERROR 0x0200: Receive screen size error LIB_AV_MASK_ TVMEMO_OFF 0x0400: Notify completion of replaying response message ELIB_AV_MASK_FLIP_SURF 0x0800: Notify completion of setting synchronous display ELIB_AV_MASK_ALL 0xffffffff: Notify all
func	MsbFunc	I	Callback function
Return value	Type	I/O	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (message sending failure)
Remark			
Videophone state notification event structures			

```
typedef struct ELIB_AV_EVENT{   /* Videophone application communication event structure */
  int   category; /* Category                    */
  int   subtype;   /* Videophone state (message name)        */
  int   info;      /* Event additional information 1             */
  int   subinfo;   /* Event additional information 2              */
  union {
    _ELIB_AV_CHANNEL_STATE   chan_state_ptr;
    _ELIB_AV_CHANNEL_START   chan_start_ptr;
    _ELIB_AV_FSTAT_CHANGE    fstat_change_ptr;
    _ELIB_AV_SETSURF         setsurf_ptr;
  }data;
 } _ELIB_AV_EVENT;
```

Completion state

```
  ELIB_AV_OK          /* Success */
  ELIB_AV_NG          /* Failure */
```

Videophone state (Message name: subtype)

1. Acknowledge videophone start (AVNotify_START_CONF) --- State in which 3G324M negotiation is completed.
   After this event, the connection being made screen shall be entered.
   If this fails, both the ELIB videophone service and 3G324M shall be ended.

2. Notify videophone start (AVNotify_START_IND) --- State in which communication is in progress.
   This message shall be notified when the open processing for all the channels is completed, regardless of success or failure.

3. Notify send/receive path state (AVNotify_PASS_STATE) --- This message shall be notified if the state of the AV send/receive channel is changed.
   The image being prepared state shall be entered when the first one at the start of videophone communication is received.

4. Notify completion of setting overlay (AVNotify_START_CONF) --- State in which the setting of overlay is completed.
   Until this notification is received, the next overlay setting cannot be made.
   The graphic display shall be updated after this notification is received.

5. Notify communication state (AVNotify_COMM_STATE) --- Notifies the circuit and channel states encountered while videophone communication is in progress.
   - One-screen decoding completion --- Notified when one screen of video data is received from the remote subscriber after a video receive channel is opened. At this point in time, the image from the remote subscriber can start to be displayed.
   - Voice data receive start --- Notified when audio data starts to be received from the remote subscriber after an audio receive channel is opened.
   - Communication error --- Notified when a circuit quality deterioration (serious) or a control circuit error is received from 3G324M. When this notification is received, an instruction to end videophone communication shall be issued.

6. Notify videophone end (AVNotify_END_IND) --- Notifies that videophone communication is ended.
   The connection time display shall start (a common procedure used whether the local subscriber or the remote subscriber disconnects).
   When this message is received, the end processing for call control shall be performed.

7. Notify completion of setting image to send to remote subscriber (AVNotify_VIDEO_SND_IND) --- Completion state (success): Normal end of both the bullet and flip processing for an image to be sent to the remote subscriber

An overlay setting including this surface is enabled.
Completion state (failure): Failure in either the bullet or flip processing. The image cannot be displayed.

8. Notify change of housing state during videophone communication (AVNotify_FLIP_IND) --- State in which the housing state (flip open or closed) is changed during videophone communication.
   The videophone ELIB shall perform the following processing:
   (1) Flip closed: If the flip closed is detected, videophone ELIB shall read the operation to be performed after closing (disconnection, call hold, or mute) from the involatile memory, sets parameters together with the closed state, and notify the application using this message (Elib shall convert the after-closing operation as required).
   (2) Flip open: If the flip open is detected, videophone ELIB shall read the operation to be performed after closing (disconnection, call hold, or mute) from the involatile memory, sets parameters together with the open state, and notify the application using this message (Elib shall convert the after-closing operation as required).

9. Receive screen size error (AVNotify_PSIZE_ERR) --- Notified when the screen received from the remote subscriber is not QCIF or SQCIF.

10. Notify completion of response message (AVNotify_TVMEMO_OFF) --- Notified when the replay of a videophone message memo response message is completed.

11. Notify completion of setting synchronous display (AVNotify_TIMING_IND) --- Notifies that the flip of the primary surface is completed and the overlay setting is enabled.

Information element information structure

- Notify send/receive path state event structure

```
typedef struct ELIB_AV_CHANNEL_STATE{
  int  AudioSnd;  /* Audio send channel state */
  int  AudioRcv;  /* Audio receive channel state */
  int  VideoSnd;  /* Video send channel state */
  int  VideoRcv;  /* Video receive channel state */
} _ELIB_AV_CHANNEL_STATE
```

Channel state

```
ELIB_AV_CHANNEL_OPEN     /* Open */
ELIB_AV_CHANNEL_CLOSE    /* Not open */
ELIB_AV_CHANNEL_FAILURE  /* Open failure (Failed to open during initialization) */
```

- Notify communication state event structure

```
typedef struct ELIB_AV_CHANNEL_STAR{
  int  kind;   /* Communication state notification type */
  int  width;  /* Width */
  int  height; /* Height */

} _ELIB_AV_CHANNEL_START
```

Communication state notification type

```
ELIB_AV_CHANNEL_STATE_1STVIDEODATA  /* One-screen decoding completion */
ELIB_AV_CHANNEL_STATE_1STAUDIODATA  /* First data reception (voice) */
ELIB_AV_CHANNEL_STATE_ERROR         /* Communication error */
```

Except for the one-screen decode completion, the width and height shall be set to Don't Care.

- Notify change of housing state during videophone communication

```
typedef struct ELIB_AV_FSTAT_CHANGE{
  int  Mode;    /* Housing open or closed state */
  int  Action;  /* Operation after closing */
```

```
  } _ELIB_AV_FSTAT_CHANGE

Housing open or closed state
 ELIB_AV_FLIP_OPEN    /* Housing open */
 ELIB_AV_FLIP_CLOSE   /* Housing closed */

Operation after closing
 ELIB_AV_FLCLOSE_DISC   /* Disconnection */
 ELIB_AV_FLCLOSE_HOLD   /* Call hold */
 ELIB_AV_FLCLOSE_MUTE   /* Mute */

- Notify completion of setting image to send to remote subscriber event structure
 typedef struct ELIB_AV_SETSURF{
   int   Type;    /* Display type */   * For information on display types, see Section A-9, "Set image to send to remote subscriber."
 } _ELIB_AV_SETSURF
```

*1 If one application repetitiously calls the start function, the mask pattern specified for the last call shall be effective. If multiple events need to be requested, the mask patterns shall be ORed.

3.8 Cancel event generation notification

Classification	Videophone ELIB function		
Function	Cancel event generation notification	Symbol	Elib_AV_Cancel
Functional overview			

Use conditions:
No limitation
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
Inhibit the transmission of any notification of an event specified by the application that called this function.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Cancel (Ap_ID, Mask)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mask	unsigned int	I	Masks for events for which notification should be ended Enabled if the bit is ON and disabled if it is OFF ELIB_AV_MASK_TVSTART 0x0001: Acknowledge videophone start ELIB_AV_MASK_TVTLK 0x0002: Notify videophone start ELIB_AV_MASK_TVEND 0x0004: Notify videophone end ELIB_AV_MASK_CHANNEL_PATH 0x0008: Notify send/receive path state ELIB_AV_MASK_TVSTATUS 0x0010: Notify communication state ELIB_AV_MASK_OVERLAY 0x0020: Notify completion of setting overlay ELIB_AV_MASK_SETSURF 0x0040: Notify completion of setting image to send to remote office ELIB_AV_MASK_FSTAT_CHG 0x01000: Notify change of housing state during videophone communication ELIB_AV_MASK_RSIZE_ERROR 0x0200: Receive screen size error ELIB_AV_MASK_ TVMEMO_OFF 0x0400: Notify completion of replaying response message ELIB_AV_MASK_ FRIP_SURF 0x0800: Notify completion of setting synchronous display ELIB_AV_MASK_ALL 0xffffffff: Notify all
Return value	Type	I/O	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (message sending failure)
Remark			

3.9 Get state of sending data to remote office

Classification	Videophone ELIB function		
Function	Get state of sending data to remote office	Symbol	Elib_AV_Get_SendData_State
Functional overview			
The state of send data to the remote office shall change depending on when a send start instruction is issued to 3G324M.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_SendData_State (Ap_ID,State)		

Argument	Type	I/O	Description
Ap_ID	unsigned int*	I	Application ID

State	int*	O	State of send data to remote office ELIB_AV_DISPTYPE_OFF /* Not sent */ ELIB_AV_ DISPTYPE _CAMERA /* Self-image */ ELIB_AV_ DISPTYPE _SUBPICTURE /* Alternate image */ ELIB_AV_ DISPTYPE _ RCVHOLD /* Response hold image */ ELIB_AV_ DISPTYPE _TALKHOLD /* Call hold image */ ELIB_AV_ DISPTYPE _MOVIE /* Movie file */ ELIB_AV_ DISPTYPE _TVMEMO /* Videophone message memo image */ ELIB_AV_DISPTYPE_CAMM_NG /* Camera out of order image */ ELIB_AV_DISPTYPE_VOICEMEMO /* Videophone voice message */ ELIB_AV_DISPTYPE_PICTURE /* Static image file */ ELIB_AV_DISPTYPE_VCHNG 0x80000000 /* Sending disabled because of a failure in opening Vch */ *1 If an image cannot be sent because of a failure in opening Vch, perform the following operation: For an alternate image: ELIB_AV_DISPTYPE_VCHNG | ELIB_AV_DISPTYPE_SUBPICTURE

Return value	Type	-	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading) */

Remark

3.10 Get send/receive path state

Classification	Videophone ELIB function			
Function	Get send/receive path state		Symbol	Elib_AV_Get_Channel_State
Functional overview				
Get the state of the channel. This function checks whether the channel is open or closed, regardless of the data receive state.				
Include file	srv_av.h			
Calling sequence	int Elib_AV_Get_Channel_State (Ap_ID)			
Argument	Type	I/O	Description	
Ap_ID	Unsigned int	I	Application ID	
AudioSnd	int*	O	Audio send channel state	
AudioRcv	int*	O	Audio receive channel state	
VideoSnd	int*	O	Audio send channel state	
VideoRcv	int*	O	Video receive channel state	
Return value	Type	-	Description	
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading)	
Remark				
/* Channel state */ ELIB_AV_CHANNEL_OPEN /* Open */ ELIB_AV_CHANNEL_CLOSE /* Not open */ ELIB_AV_CHANNEL_FAILURE /* Open failure */ /* (Failed to open during initialization) */				

3.11 Send videophone DTMF

Classification	Videophone ELIB function		
Function	Send Videophone DTMF	Symbol	Elib_AV_Send_DTMF
Functional overview			
Use conditions: Videophone communication is in progress. Send message to the videophone ELIB thread: Notify sending videophone DTMF Videophone ELIB processing: Videophone ELIB shall control the sounding of DTMF tone.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Send_DTMF (Ap_ID, DtmfList)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
DtmfList	Unsigned char*	I	An array that stores key information that has been entered. Only one-digit key information can be entered. The key information shall be ASCII code. An array shall be terminated with NULL. The following 12 keys are valid as DTMF. "0,1,2,3,4,5,6,7,8,9,#,* "
Return value	Type	I/O	Description
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Parameter error, message sending failure)
Remark			
- If this function is repetitiously called, up to 256 bytes can be stored. - If this function is called when the stored key information is at its upper limit, the data shall be abandoned and the value OK shall be returned. - If sending fails, the data shall be abandoned.			

3.12 Synchronous display

Classification	Videophone ELIB function		
Function	Synchronous display	Symbol	Elib_AV_Flip_Surf

Functional overview

Use conditions:

- This function shall be called when the setting of an image to send to the remote office is completed (when a notification of completion of setting an image to send to the remote office is received) and the overlay setting is completed (a notification of completion of setting an overlay is received).
- After this function is called, this function shall not be called again until a notification of completion of setting synchronous display is received.

Send message to the videophone ELIB thread:

Instruction to set synchronous display

Videophone ELIB processing:

Perform the flip processing of the primary surface and enable the overlay setting.

The completion of synchronous display shall be notified using a message notifying the completion of synchronous display.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Flip_Surf (AP_ID)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	I/O	Description
Ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (message sending failure)

Remark

* This function, when called, shall perform the control of event flag [wai_flgx] and put the videophone APL in the wait state.
(When the wait state is cleared, clr_flgx shall be used to clear the event flag.)

3.13 End videophone

Classification	Videophone ELIB function		
Function	Close videophone	Symbol	Elib_AV_Close_TV
Functional overview	Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: None		
Include file	srv_av.h		
Calling sequence	int Elib_AV_Close_TV(AP_ID)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	I/O	Description
Ret	Int	O	Normal end: ELIB_AV_OK
Remark	- If this function is called, no processing shall be performed and the value "OK" shall be returned. * This is because the sharing of audio resources between the movie ELIB and the videophone ELIB is supported by the audio side, eliminating the need of calling this function.		

3.14 Set static image file image

Classification	AV setting service functions		
Function	Set static image file image	Symbol	Elib_AV_Set_Send_PictureFile
Functional overview			

Use conditions:
Set the image for replaying a static image file.
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
Set the image for replaying a static image file.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_Send_PictureFile (AP_ID, Imginfo[3])		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
ImgInfo[3]	unsigned char *	I	Pointer to the image display structure
Return value	Type	-	Description
Ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Parameter error, message sending failure)

Remark

Detailed explanation of an image display structure (ImgInfo)
/* Variable for registering and setting image display data */
unsigned char* ImgInfo[3];
The following lists pointers to YcbCr data.
ImgInfo[0] Initial address of Y data storage area (25344 bytes)
ImgInfo[1] Initial address of Cb data storage area (6336 bytes)
ImgInfo[2] Initial address of Cr data storage area (6336 bytes)

[Reference]
The numbers of bytes (sizes) listed above are defined in srv_av.h as the #define value as shown below:

```
#define ELIB_AV_YCBCR_QCIF_Y_SIZE 25344 /* Y component size of QCIF size Y/Cb/Cr image */
#define ELIB_AV_YCBCR_QCIF_CB_SIZE 6336 /* Cb component size of QCIF size Y/Cb/Cr image */
#define ELIB_AV_YCBCR_QCIF_CR_SIZE 6336 /* Cr component size of QCIF size Y/Cb/Cr image */
```

- The replay of a static image file requires:
  (1) Set overlay (synchronous flip)
  (2) This function
  (3) Set image to send to remote office (static image file)
  (4) Calling the four functions for synchronous display in a series of processing (without waiting for a completion notification from ELIB)
- If this function is repetitiously called, the value NG shall be returned.
- The memory acquired by the application for image data may be released when this function returns.
- The image data passed by this function to the videophone ELIB shall be effective until the next instruction to set an image to send to a remote office is issued and, at this point in time, the image data in ELIB shall be abandoned.

3.15 Stop sending videophone DTMF

Classification	Videophone ELIB function		
Function	Stop sending videophone DTMF	Symbol	Elib_AV_Stop_DTMF
Functional overview			

Use conditions:
Clear the data in the DTMF data queue and, if the Audio API is starting the sounding of DTMF, wait for the completion of sounding.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Stop_DTMF (AP_ID)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Return value	Type	I/O	Description
Ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (message sending failure)

Remark

- This function shall be used:
(1) Before a voice memo or movie file is replayed
(2) If the flip close operation occurs while DTMF is being sent (when the close operation is set to hold or mute and the earphone is unplugged)

[The following is a matter to be decided]
* This function, when called, shall perform the control of event flag [wai_flgx] and put the videophone APL in the wait state.
(When the wait state is cleared, clr_flgx shall be used to clear the event flag.)

3.16 Select and set alternate image

Classification	AV setting service functions		
Function	Select and set alternate image	Symbol	Elib_AV_Set_SubPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) B- 4 Select and set an alternate static image to be sent when sending of self image upon call origination is disabled.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_SubPicture(Ap_ID, Mode, AvatarFileID)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_SUBST_FIX: Fixed (preinstalled) image ELIB_AV_SUBST_USER: User image
AvatarFileID	unsigned int	I	Avatar file ID
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			
No avatar file ID shall be used. (because the specifications have no avatar function.)			

3.17 Get setting state of selecting alternate image

Classification	AV setting service functions		
Function	Get setting state of selecting alternate image	Symbol	Elib_AV_Get_SubPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) When displaying an alternate image, get the image reference address of the alternate image.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_SubPicture(Ap_ID , *Mode, AvatarFileID)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
AvatarFileID	unsigned int*	O	Avatar file ID
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			
No avatar file ID shall be used.(because the specifications have no avatar function.)			

3.18 Set sending of self image upon call origination

Classification	AV setting service functions		
Function	Set sending of self image upon call origination	Symbol	Elib_AV_Set_Send

Functional overview

Use conditions:
None
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
(1) Set whether to send a self image (enabling the sending of a self image upon call origination) or not (disabling the sending of a self image upon call origination) regardless of the ON/OFF state of the camera when an originated videophone call is received by a remote subscriber, initiating videophone communication.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_Send(Ap_ID , Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_SEND_ON: Enabling the sending of self image upon call origination ELIB_AV_SEND_OFF: Disabling the sending of self image upon call origination
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)

Remark

3.19 Get setting state of sending self image upon call origination

Classification	AV setting service functions		
Function	Get setting state of sending self image upon call origination	Symbol	Elib_AV_Get_Send
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of sending a self image upon call origination.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_Send(Ap_ID , *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_SEND_ON: Enabling the sending of self image upon call origination ELIB_AV_SEND_OFF: Disabling the sending of self image upon call origination
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading)
Remark			

3.20 Set communication mode

Classification	AV setting service functions		
Function	Set communication mode	Symbol	Elib_AV_Set_Picture_Quality

Functional overview

Use conditions:
None
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
(1) Set the image quality.
(2) Start communication with the specified image quality.
Calling this function during videophone communication will not switch communication modes (During communication, switch communication modes using A-5).

There shall be three image quality settings: Movement priority, standard, and quality priority.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_Picture_Quality(Ap_ID , Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	Image quality mode ELIB_AV_VIDEO_MOVIE: Movement priority ELIB_AV_VIDEO_NORMAL: Standard ELIB_AV_VIDEO_QUALITY: Quality priority
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)

Remark

Image quality mode
- Default: Standard
- The setting value shall be retained after the power is turned off.

3.21 Get setting state of communication mode

Classification	AV setting service functions		
Function	Get setting state of communication mode	Symbol	Elib_AV_Get_Picture_Quality
Functional overview			

Use conditions:
None
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
(1) Get the setting state of the image quality specified in the communication mode setting.
(This is not the setting due to switching of communication modes during videophone communication.)

Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_Picture_Quality(Ap_ID , *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	Image quality mode ELIB_AV_VIDEO_MOVIE: Movement priority ELIB_AV_VIDEO_NORMAL: Standard ELIB_AV_VIDEO_QUALITY: Quality priority
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading)

Remark

3.22 Set remote surveillance

Classification	AV setting service functions		
Function	Set remote surveillance	Symbol	Elib_AV_Set_Remote_Surv
Functional overview			

Use conditions:
None
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
(1) Set whether the remote surveillance shall monitor a videophone call when it is received.

Switch between ON (Response time from call reception to response: 3 to 120 seconds) and OFF using a password.
Up to five remote subscriber numbers for automatic response can be set.

Enabling this function while the manner mode or automatic receiving setting is enabled will disable the manner mode or automatic receiving setting.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_Remote_Surv(Ap_ID ,Type,Mode,Time,EntryNo,*DialNo)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Type	int	I	Setting type - For a mode setting, only the setting mode parameter shall be valid. - For a surveillance time setting, only the surveillance time parameter shall be valid. - For a registration number setting, the registration number and the dial number to be registered shall be valid.
Mode	int	I	Setting mode
Time	int	I	Surveillance time (Specify time from 3 to 120 seconds.)
EntryNo	int	I	Registration number
DialNo	unsigned char*	I	Dial number to be registered ELIB_AV_REMOTE_SURV_DIALMAX: 26 /* Maximum number of digits for telephone number */ The dial information shall be ASCII code terminated with NULL.
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

```
/* Setting type */
  ELIB_AV_REMOTE_SETMODE: Mode setting
  ELIB_AV_REMOTE_SETTIME: Surveillance time setting
  ELIB_AV_REMOTE_SETDIALNO: Registration number setting

/* Setting mode */
  ELIB_AV_REMOTE_ON: Remote surveillance enabled
  ELIB_AV_REMOTE_OFF: Remote surveillance disabled

/* Registration number */
  ELIB_AV_ENTRY_NO1: Registration number 1
  ELIB_AV_ENTRY_NO2: Registration number 2
  ELIB_AV_ENTRY_NO3: Registration number 3
  ELIB_AV_ENTRY_NO4: Registration number 4
  ELIB_AV_ENTRY_NO5: Registration number 5
```

3.23 Get setting state of remote surveillance

Classification	AV setting service functions		
Function	Get setting state of remote surveillance	Symbol	Elib_AV_Get_Remote_Surv

Functional overview

Use conditions:
None
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
(1) Get the setting state of remote surveillance.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_Remote_Surv(Ap_ID ,*Mode,*Time,*DialNo)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	Setting mode
Time	int*	O	Surveillance time (Specify time from 3 to 120 seconds.)
DialNo[][ELIB_AV_REMOTE_SURV_DIALMAX+1]	unsigned char*	O	Dial number to be registered DialNo[ELIB_AV_REMOTE_ENTRYMAX][ELIB_AV_REMOTE_SURV_DIALMAX+1] ELIB_AV_REMOTE_ENTRYMAX: 5 /* Maximum number of registration numbers */ ELIB_AV_REMOTE_SURV_DIALMAX: 26 /* Maximum number of digits for telephone number */
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading)

Remark

/* Setting mode */
ELIB_AV_REMOTE_ON: Remote surveillance enabled
ELIB_AV_REMOTE_OFF: Remote surveillance disabled

3.24 Get setting state of receive denial of videophone

Classification	AV setting service functions		
Function	Get setting state of receive denial of videophone	Symbol	Elib_AV_Get_Rcv_Denial

Functional overview
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of receive denial of videophone.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_Rcv_Denial(Ap_ID ,*Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_RCVDENIAL_ON: Receive denial enabled ELIB_AV_RCVDENIAL_OFF: Receive denial enabled
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading)

Remark

3.25 Set videophone message memo image

Classification	AV setting service functions		
Function	Set videophone message memo image	Symbol	Elib_AV_Set_MessageMemoPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Select and set an image to be sent to the remote subscriber during videophone message memo recording when a videophone call is received.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_MessageMemoPicture(Ap_ID ,Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			
This function shall be used to set a "Message memo being recorded" image.			

3.26 Get setting state of videophone message memo image

Classification	AV setting service functions		
Function	Get setting state of videophone message memo image	Symbol	Elib_AV_Get_MessageMemoPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of a "Videophone message memo being recorded" image.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_MessageMemoPicture(Ap_ID , *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			
This function shall be used to get the setting state of a "Message memo being recorded" image.			

3.27 Set call hold image

Classification	AV setting service functions		
Function	Set call hold image	Symbol	Elib_AV_Set_TalkHoldPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Set an image to be sent to the remote subscriber when the call is put on hold during videophone communication.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_TalkHoldPicture(Ap_ID , Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.28 Get setting state of call hold image

Classification	AV setting service functions		
Function	Get setting state of call hold image	Symbol	Elib_AV_Get_TalkHoldPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of the call hold image.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_TalkHoldPicture(Ap_ID , *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.29 Set response hold image

Classification	AV setting service functions		
Function	Set response hold image	Symbol	Elib_AV_Set_RcvHoldPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Set an image to be sent to the remote subscriber during response hold when a videophone call is received.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_RcvHoldPicture(Ap_ID , Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.30 Get setting state of response hold image

Classification	AV setting service functions		
Function	Get setting state of response hold image	Symbol	Elib_AV_Get_RcvHoldPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of the response hold image.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_RcvHoldPicture(Ap_ID , *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.31 Select and set 64K/32K call origination

Classification	AV setting service functions		
Function	Select and set 64K/32K call origination	Symbol	Elib_AV_Set_Rate
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Select and set 64K or 32K call origination.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_Rate(Ap_ID , Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_RATE_64K: 64K call origination ELIB_AV_RATE_32K: 32K call origination
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.32 Get setting state of selecting 64K/32K call origination

Classification	AV setting service functions		
Function	Select and set 64K/32K call origination	Symbol	Elib_AV_Get_Rate
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of selecting 64K or 32K call origination.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_Rate(Ap_ID , *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_RATE_64K: 64K call origination ELIB_AV_RATE_32K: 32K call origination
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading)
Remark			

3.33 Set fallback

Classification	AV setting service functions		
Function	Set fallback	Symbol	Elib_AV_Set_FallBack_32k
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Perform the logging processing and return NG.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_FallBack_32k(Ap_ID, Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_FALLBACK_32K_ON: Fallback enabled ELIB_AV_FALLBACK_32K_OFF: Fallback disabled
Return value	Type	-	Description
ret	int	O	Abnormal end: ELIB_AV_NG (parameter error)
Remark			
If this function is called, the value NG shall be returned (No processing).			

3.34 Get setting state of fallback

Classification	AV setting service functions		
Function	Get setting state of fallback	Symbol	Elib_AV_Get_FallBack_32k
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Perform the logging processing and then return NG.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_FallBack_32k(Ap_ID , *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_FALLBACK_32K_ON: Fallback enabled ELIB_AV_FALLBACK_32K_OFF: Fallback disabled
Return value	Type	-	Description
ret	int	O	Abnormal end: ELIB_AV_NG (Failure in reading)
Remark			
If this function is called, the value NG shall be returned (No processing).			

3.35 Set fallback to voice

Classification	AV setting service functions		
Function	Set fallback to voice	Symbol	Elib_AV_Set_FallBack_Voice
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Set whether to originate a call on voice communications if a call on videophone at 32k is originated and then disconnected.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_FallBack_Voice (Ap_ID , Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_FALLBACK_VOICE_ON: Fallback to voice enabled ELIB_AV_FALLBACK_VOICE_OFF: Fallback to voice disabled
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.36 Get setting state of fallback to voice

Classification	AV setting service functions			
Function	Get setting state of fallback to voice		Symbol	Elib_AV_Get_FallBack_Voice
Functional overview				
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of fallback to voice.				
Include file	srv_av.h			
Calling sequence	int Elib_AV_Get_FallBack_Voice (Ap_ID , *Mode)			
Argument	Type	I/O	Description	
Ap_ID	unsigned int	I	Application ID	
Mode	int*	O	ELIB_AV_FALLBACK_VOICE_ON: Fallback to voice enabled ELIB_AV_FALLBACK_VOICE_OFF: Fallback to voice disabled	
Return value	Type	-	Description	
ret	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading)	
Remark				

3.37 Set switching of parent-child screens

Classification	AV setting service functions		
Function	Set switching of parent-child screens	Symbol	Elib_AV_Set_MainDisp_Change
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Set the switching of parent-child screens and the sending of the local subscriber image.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_MainDisp_Change (Ap_ID , Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	Setting of parent-child screens ELIB_AV_MODE1: Parent screen: Remote subscriber image, Child screen: Local subscriber camera ELIB_AV_MODE2: Parent screen: Remote subscriber image, Child screen: Local subscriber alternate image ELIB_AV_MODE3: Parent screen: Remote subscriber image, Child screen: No local subscriber image ELIB_AV_MODE4: Parent screen: Local subscriber camera, Child screen: Remote subscriber image
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.38 Get setting state of switching of parent-child screens

Classification	AV setting service functions		
Function	Get setting state of switching of parent-child screens	Symbol	Elib_AV_Get_MainDisp_Change

Functional overview

Use conditions:
None
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
(1) Get the setting state of switching of parent-child screens.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_MainDisp_Change (Ap_ID , *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	Setting of parent-child screens ELIB_AV_MODE1: Parent screen: Remote subscriber image, Child screen: Local subscriber camera ELIB_AV_MODE2: Parent screen: Remote subscriber image, Child screen: Local subscriber alternate image ELIB_AV_MODE3: Parent screen: Remote subscriber image, Child screen: No local subscriber image ELIB_AV_MODE4: Parent screen: Local subscriber camera, Child screen: Remote subscriber image ELIB_AV_MODE5: Parent screen: Local subscriber alternate image, Child screen: Remote subscriber image ELIB_AV_MODE6: Parent screen: Local subscriber camera, Child screen: No remote subscriber image ELIB_AV_MODE7: Parent screen: Local subscriber alternate image, Child screen: No remote subscriber image
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (Failure in reading)

Remark

3.39 Set image display size

Classification	AV setting service functions		
Function	Set image display size	Symbol	Elib_AV_Set_Picture_DispSize
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Set the parent screen image display size to be used during videophone communication.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_Picture_DispSize(Ap_ID, Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_SIZE_MAINDISP: Screen size ELIB_AV_SIZE_REAL: Real size
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.40 Get setting state of image display size

Classification	AV setting service functions		
Function	Get setting state of image display size	Symbol	Elib_AV_Get_Picture_DispSize
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of the parent screen image display size to be used during videophone communication.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_Picture_DispSize(Ap_ID, *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_SIZE_MAINDISP: Screen size ELIB_AV_SIZE_REAL: Real size
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.41 Set videophone message replay image

Classification	AV setting service functions		
Function	Set videophone message replay image	Symbol	Elib_AV_Set_MessagePlayPicture

Functional overview

Use conditions:
None
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
(1) Select and set an image to be sent to the remote subscriber during videophone message memo response when a videophone call is received.

Include file	srv_av.h		
Calling sequence	int Elib_AV_Set_MessagePlayPicture(Ap_ID,Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int	I	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
Return value	Type	-	Description
Ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)

Remark

3.42 Get setting state of videophone message memo replay image

Classification	AV setting service functions		
Function	Get setting state of videophone message memo replay image	Symbol	Elib_AV_Get_MessagePlayPicture
Functional overview			
Use conditions: None Send message to the videophone ELIB thread: None Videophone ELIB processing: (1) Get the setting state of the videophone message memo replay image.			
Include file	srv_av.h		
Calling sequence	int Elib_AV_Get_MessagePlayPicture(Ap_ID, *Mode)		
Argument	Type	I/O	Description
Ap_ID	unsigned int	I	Application ID
Mode	int*	O	ELIB_AV_SUBST_FIX: Fixed image ELIB_AV_SUBST_USER: User image
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG (parameter error)
Remark			

3.43 Request for registering image display data

Classification	AV image service functions		
Function	Request for registering image display data	Symbol	Elib_AV_DataSave
Functional overview			

Use conditions:
The data to be registered shall be Y/Cb/Cr format and fixed to QCIF size.
Send message to the videophone ELIB thread:
None
Videophone ELIB processing:
(1) Register image data.

Include file	srv_av.h		
Calling sequence	int Elib_AV_DataSave(int Ap_ID, int ImgNo, unsigned char * ImgInfo[3], _ELIB_AV_UIM_INF *Uim_Inf);		
Argument	Type	I/O	Description
Ap_ID	Int	I	Application ID
ImgNo	Int	I	Registered image type ELIB_AV_NUMBER02 /* Alternate image */ ELIB_AV_NUMBER04 /* Message memo (record) image */ ELIB_AV_NUMBER06 /* Call hold image */ ELIB_AV_NUMBER08 /* Response hold image */ ELIB_AV_NUMBER13 /* Message memo (replay) image */
ImgInfo[3]	unsigned char *	I	Pointer to the image display structure
Uim_Inf	_ELIB_AV_UIM_INF *	I	Pointer to the UIM identification information structure
Return value	Type	-	Description
re,"	Int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG Parameter error: ELIB_AV_PARAM_NG Failure in writing: ELIB_AV_WRITENG

Remark

- Detailed explanation of an image display structure (ImgInfo)

/* Variable for registering and setting image display data */ *1

unsigned char* ImgInfo[3];

The following lists pointers to YCbCr data.

ImgInfo[0] Initial address of Y data storage area (25344 bytes)
ImgInfo[1] Initial address of Cb data storage area (6336 bytes)
ImgInfo[2] Initial address of Cr data storage area (6336 bytes)

[Reference]
The numbers of bytes (sizes) listed above are defined in srv_av.h as the #define value as shown below:

```
#define ELIB_AV_YCBCR_QCIF_Y_SIZE 25344 /* Y component size of QCIF size Y/Cb/Cr image */
#define ELIB_AV_YCBCR_QCIF_CB_SIZE 6336 /* Cb component size of QCIF size Y/Cb/Cr image */
#define ELIB_AV_YCBCR_QCIF_CR_SIZE 6336 /* Cr component size of QCIF size Y/Cb/Cr image */
```

- Detailed explanation of a UIM identification information structure

```
typedef struct ELIB_AV_UIM_INF {
    unsigned int      UimCheck; /* UIM check identifier          */
    _ELIB_DRM_UIM_ID  Uim_ID;           /* UIM identification ID information structure    */
} _ELIB_AV_UIM_INF;                     /* UIM identification information structure       */
```

* The UIM check identifier shall be set to either "Check enabled: ELIB_AV_UIM_ON" or "Check disabled: ELIB_AV_UIM_OFF."

*1 For the variable for registering and setting image display data, memory area shall be allocated using the memory allocation function in advance.

3.44 Request for getting image display data

Classification	AV image service functions		
Function	Request for registering image display data	Symbol	Elib_AV_DataRead
Functional overview			

Use conditions:

Before calling this function, allocate an image data storage area using the memory allocation function (YCbCr format, fixed to QCIF).

Send message to the videophone ELIB thread:

None

Videophone ELIB processing:

(1) Get image display data.

(2) Return the default image if no user image is registered yet or the file API returns an error.

* Specifying an image

In the image specification (ImgNo), specify an image ID (index) in the target image header.

Include file	srv_av.h		
Calling sequence	int Elib_AV_DataRead(int Ap_ID , int ImgNo, unsigned char* ImgInfo[3]);		
Argument	Type	I/O	Description
Ap_ID	int	I	Application ID
ImgNo	int	I	Image data storage number ELIB_AV_NUMBER01 /* Alternate image, fixed */ ELIB_AV_NUMBER02 /* Alternate image, user */ ELIB_AV_NUMBER03 /* Message memo (record) image, fixed */ ELIB_AV_NUMBER04 /* Message memo (record) image, user */ ELIB_AV_NUMBER05 /* Call hold image, fixed */ ELIB_AV_NUMBER06 /* Call hold image, user */ ELIB_AV_NUMBER07 /* Response hold image, fixed */ ELIB_AV_NUMBER08 /* Response hold image, user */ ELIB_AV_NUMBER12 /* Message memo (replay) image, fixed */ ELIB_AV_NUMBER13 /* Message memo (replay) image, user */
ImgInfo[3]	unsigned char *	O	Pointer to the storage location of the image data structure See the detailed explanation on an image display structure in C-1, "Request for registering image display data."
Return value	Type	-	Description
ret	int	O	Normal end: ELIB_AV_OK Abnormal end: ELIB_AV_NG

			Parameter error: ELIB_AV_PARAM_NG
Remark			